Efficient Implementation of a Generalized Pair Hidden Markov Model for Comparative Gene Finding

William H. Majoros, Mihaela Pertea, Steven L. Salzberg
Bioinformatics Department, The Institute For Genomic Research,
Rockville, MD, USA

Abstract

The increased availability of genome sequences of closely related organisms has generated much interest in utilizing homology to improve the accuracy of gene prediction programs. Generalized Pair Hidden Markov Models (GPHMMs) have been proposed as one means to address this need. However, all GPHMM implementations currently available are either closed-source or the details of their operation are not fully described in the literature, leaving a significant hurdle for others wishing to advance the state of the art in GPHMM design. We have developed an open-source GPHMM gene finder, TWAIN, which performs very well on two related *Aspergillus* species, *A. fumigatus* and *A. nidulans*, finding 89% of the exons and predicting 74% of the gene models exactly correctly in a test set of 147 conserved gene pairs. We describe the implementation of this GPHMM and we explicitly address the assumptions and limitations of the system. We suggest possible ways of relaxing those assumptions to improve the utility of the system without sacrificing efficiency beyond what is practical.

Availability : Available under the open-source Artistic License at:
http://www.tigr.org/software/pirate/twain/twain.html

Contact : bmajoros@tigr.org

Introduction

As the amount of genomic sequence available in public archives skyrockets, our reliance on purely or largely automatic methods of identifying genes in unannotated sequence will likely also increase. Unfortunately, *ab initio* methods of gene prediction are far from perfect. This has prompted some researchers in the field to investigate the use of homology evidence to improve the accuracy of current gene-finding technology. Fortunately, for a growing number of species, genome centers are now generating two or more sequences from closely related organisms. The most well-known (and largest) related species are the human, mouse, and rat genomes, soon to be joined by the chimpanzee and other mammals. Less well-known but equally useful are a large set of related *Drosophila* species (12 in all), several fungal genomes, at least four yeast genomes (Kellis *et al.*, 2003), three trypanosomids, and others soon to come. This increased availability of genome sequences from closely related organisms has generated much interest of late in utilizing apparent synteny to improve the accuracy of gene prediction programs. The idea behind this strategy is that the patterns of conservation between close relatives at the level of amino acids versus nucleotides should follow the corresponding transitions between coding and noncoding regions within those genomes. Generalized Pair Hidden Markov Models (GPHMMs) have been proposed as one means to observe these patterns of conservation and incorporate them into the gene prediction process (Pachter *et al.*, 2002; Alexandersson *et al.*, 2003). GPHMMs provide an attractive theoretical framework for comparative gene finding, and constitute a natural progression from earlier *ad hoc* methods (e.g., Bafna and Huson, 2000; Novichkov, 2001).

Though at most a handful of Generalized Pair HMMs for gene finding have been published to date (e.g., Alexandersson *et al.*, 2003), detailed methods for their efficient implementation have not been fully described, making it difficult for others to replicate or extend the research that has so far been reported. GPHMM gene finders are especially difficult to implement efficiently due to the combinatorial explosion inherent in considering all possible pairs of open reading frames in two genomes. A naïve implementation of a GPHMM would be useless in practice due to the enormous time and memory requirements of such an implementation (Alexandersson *et al.*, 2003).

In an attempt to help remedy this situation, we describe the algorithms and heuristics which we developed during the implementation of our open-source GPHMM gene finder, and also the assumptions that allowed us to improve the speed and memory efficiency of the program. While the program in its current form was found to achieve higher accuracy than an identically-trained non-comparative gene finder, we have hopes that even greater levels of accuracy and utility can be achieved if we can discover efficient means to relax some of the assumptions that were made in the development of our GPHMM gene finder. We suggest some possible ways in which this might be achieved.

Background

Generalized Hidden Markov Models

A number of gene-finding programs have been developed which utilize the Generalized Hidden Markov Model (GHMM) framework (e.g., Kulp *et al.*, 1996; Burge and Karlin, 1997; Stanke and Waack, 2003; Majoros *et al.*, 2004; Korf, 2004). This class of models has been found to provide competitive prediction accuracy while also providing an intuitive probabilistic interpretation of the gene-finding problem. A GHMM is a state-based generative model in which each state emits a sequence of bases comprising a feature such as an exon or intron. As traditionally formulated, a GHMM does not incorporate homology information.

Formally, let $\alpha=\{A,T,C,G\}$, $\alpha^n=\{s|s$ is a string over α of length $n\}$, $\alpha^*=\bigcup_n(\alpha^n)$, $\mathbb{N}$=the non-negative integers, and $\mathbb{R}$=the reals. We denote a GHMM as a 6-tuple $(Q,P_t,P_d,P_e,\pi_0,\pi_f)$ comprising a set of states Q with designated start state π_0 and final state π_f (both silent), a set of state transition probabilities $P_t{:}Q\times Q\rightarrow\mathbb{R}$, a set of length or "duration" probabilities $P_d{:}\mathbb{N}\times Q\rightarrow\mathbb{R}$ conditional on state, and a set of emission probabilities $P_e{:}\alpha^*\times Q\times\mathbb{N}\rightarrow\mathbb{R}$ conditional on state and duration. A single n-state run of the GHMM begins in state $q_0=\pi_0$, transitions stochastically from state q_{i-1} to state q_i for $1\leq i<n$ according to $P_t(q_i|q_{i-1})$, and terminates in state $q_{n-1}=\pi_f$. In each state q_i the

GHMM stochastically chooses duration d according to $P_d(d|q_i)$ and emits string $S_i \in \alpha^d$ according to $P_e(S_i|q_i,d)$. Note that $P_d(0|\pi_0)=P_d(0|\pi_f)=1$.

Gene-finding with a GHMM involves finding the most probable parse $\phi_{\max}$ of a given nucleotide sequence S:

$$\begin{aligned}\phi_{\max} &= \underset{\phi}{\arg\max}\, P(\phi \mid S) = \underset{\phi}{\arg\max}\, \frac{P(\phi, S)}{P(S)} \\ &= \underset{\phi}{\arg\max}\, P(\phi, S) = \underset{\phi}{\arg\max}\, P(S \mid \phi)P(\phi),\end{aligned} \tag{1}$$

where each $\phi=\{(q_i,d_i)|0\le i<n\}$ specifies a time-ordered series of states (features) and integer durations (feature lengths) during a single run of the GHMM. $P(S|\phi)$ can be factored according to the states in ϕ to produce $\prod P_e(S_i|q_i,d_i)$, where the precise formula for each $P_e(S_i|q_i,d_i)$ is defined separately for each state depending on the type of model used in the state (e.g., a Markov chain, position-specific weight matrix, etc.). $P(\phi)$ can likewise be decomposed by state into a product of transition and duration probabilities to produce:

$$\phi_{\max} = \underset{\phi}{\arg\max} \prod_{i=1}^{n-1} P_e(S_i \mid q_i, d_i) P_t(q_i \mid q_{i-1}) P_d(d_i \mid q_i)\,. \tag{2}$$

where the concatenation $S_0 \ldots S_{n-1}$ of individual features forms the input sequence S. This maximization step can be efficiently computed using a modified Viterbi decoding algorithm (Burge, 1997; Majoros *et al.*, 2005).

Let $\theta=(P_e,P_t,P_d)$ denote a particular parameterization of the GHMM. If we define the accuracy of the gene finder on a training set $T=\{(S_i,\phi_i)|0\le i<m\}$ of m sequence × parse pairs as $\sum_{(S,\phi)\in T} \log P(\phi \mid S, \theta)$, then we can expect to obtain the highest accuracy on the sequences in T when we use the GHMM parameters θ^* defined by:

$$\begin{aligned}\theta^* &= \underset{\theta}{\arg\max} \left(\sum_{(S,\phi)\in T} \log P(S,\phi \mid \theta) / P(S \mid \theta) \right) \\ &= \underset{\theta}{\arg\max} \left(\sum_{(S,\phi)\in T} \log \frac{\prod_{i=1}^{n-1} P_e(S_i \mid q_i, d_i) P_t(q_i \mid q_{i-1}) P_d(d_i \mid q_i)}{P(S \mid \theta)} \right)\end{aligned} \tag{3}$$

Because evaluation of Equation (3) is generally too expensive to perform on current commodity computing systems, the individual terms are typically optimized independently in practice (Cawley *et al.*, 2001; Rabiner, 1989), resulting in a parameterization which is not guaranteed to be globally optimal (Majoros and Salzberg, 2004) . This problem will remain when we incorporate homology into the model.

Generalized Pair Hidden Markov Models

With the increased availability of closely related genomes comes the prospect of utilizing identifiable synteny between regions of those genomes to refine the gene prediction process. An elegant method of adapting a GHMM gene finder to utilize apparent homology is by permitting each state in the GHMM to emit a pair of features, one per genome, rather than a single feature at a time. Such a model may be called a Generalized Pair Hidden Markov Model (GPHMM) (Pachter *et al.*, 2002).

Formally, we denote a GPHMM as a 6-tuple $(Q,P_t,\psi_d,\psi_e,\pi_0,\pi_f)$ in which all elements are as defined above for the GHMM except ψ_d which is a joint distribution of paired durations $\psi_d{:}\mathbb{N}\times\mathbb{N}\times Q\rightarrow\mathbb{R}$ conditional on state, and ψ_e which is a joint distribution of paired emissions $\psi_e{:}\alpha^*\times\alpha^*\times Q\times\mathbb{N}\times\mathbb{N}\rightarrow\mathbb{R}$ conditional on state and two durations. At each state q_i the GPHMM first selects a pair of durations $d_{i,1}$ and $d_{i,2}$ according to $\psi_d(d_{i,1},d_{i,2}|q_i)$ and then emits sequences $S_{i,1}$ and $S_{i,2}$ into genomes 1 and 2, respectively, according to the joint emission distribution $\psi_e(S_{i,1},S_{i,2}|q_i,d_{i,1},d_{i,2})$.

Formulated in this way, gene prediction with a GPHMM can be accomplished by finding the most probable parse $\phi_{\max}$ according to:

$$\phi_{\max} = \underset{\phi}{\arg\max} \prod_{i=1}^{n-1} \psi_e(S_{i,1},S_{i,2} \mid q_i,d_{i,1},d_{i,2})P_t(q_i \mid q_{i-1})\psi_d(d_{i,1},d_{i,2} \mid q_i) \qquad (4)$$

for features $S_{i,1}$ and $S_{i,2}$ of lengths $d_{i,1}$ and $d_{i,2}$, respectively, and where the concatenation $S_{0,j}...S_{n-1,j}$ of individual features forms the input sequence S_j, $j\in\{0,1\}$. A parse $\phi=\{(q_i,d_{i,1},d_{i,2})|0\leq i<n\}$ is now a series of states and corresponding pairs of durations. The joint emission probability term ψ_e can be decomposed via conditional probability as:

$$\psi_e(S_{i,1},S_{i,2}|q_i,d_{i,1},d_{i,2})=P_e(S_{i,1}|q_i,d_{i,1})\ P_{cond}(S_{i,2}|S_{i,1},q_i,d_{i,2}) \qquad (5)$$

(if we ignore any dependence of $S_{i,1}$ on $d_{i,2}$ and $S_{i,2}$ on $d_{i,1}$) where P_e can be estimated in the same way as for a GHMM, and P_{cond} can be estimated by aligning $S_{i,1}$ and $S_{i,2}$ and mapping the resulting alignment to a conditional probability based on the alignment properties of a set of known orthologues. Similarly, the joint duration distribution ψ_d can be estimated from known orthologues, if such are available; otherwise, $\psi_d(d_{i,1},d_{i,2}|q_i)$ may be very roughly approximated using $P_d(d_{i,1}|q_i)$ or $P_d(d_{i,2}|q_i)$, as long as we expect the durations to be highly correlated. As with the GHMM we cannot at this time recommend a simple and efficient means of obtaining the globally optimal model parameters for the GPHMM using widely available computing resources.

Note that for simplicity we have not considered GPHMMs which can emit features in one genome which are not paired with matching features in the other genome. To our knowledge, all currently available GPHMM implementations suffer from this shortcoming (though non-generalized pair HMMs exist which do allow inserted or deleted features—e.g., Doublescan (Meyer and Durbin, 2002)). We address this issue briefly in the Discussion section.

Implementation

We implemented a GPHMM-based comparative gene finder called TWAIN, which consists of three programs: MUMmer, ROSE, and OASIS (Fig. 1). Whereas OASIS implements the actual GPHMM decoding algorithm, MUMmer and ROSE act as preprocessors for OASIS, identifying regions of contigs on which to run the GPHMM decoder and also providing a set of precomputed "guide" alignments for OASIS to use as a basis for estimating P_{cond} terms. A Perl script is provided to automate the end-to-end operation of the system's pipeline.

ROSE: Region-Of-Synteny Extractor

ROSE is a program which identifies likely orthologous regions between any given pair of contigs from two genomes. These regions are then provided as inputs to the GPHMM for parsing of those regions into pairs of gene predictions.

ROSE utilizes the *MUMmer* package for whole-genome alignment (Kurtz *et al.*, 2004). The *PROmer* program distributed with this package is used to identify conserved coding regions by performing six-frame translation in both genomes and assessing local amino acid alignments between pairs of such translated regions. The result is a set of reference points on the contig pair corresponding to significant alignments in amino acid space. Similarly, *NUCmer* identifies significant HSPs (*High-scoring Segment Pairs*) in nucleotide space. These alignments are used by OASIS to approximate P_{cond} in putative coding and noncoding features, respectively, and allow the GPHMM to utilize information about both coding and noncoding conservation across the two genomes.

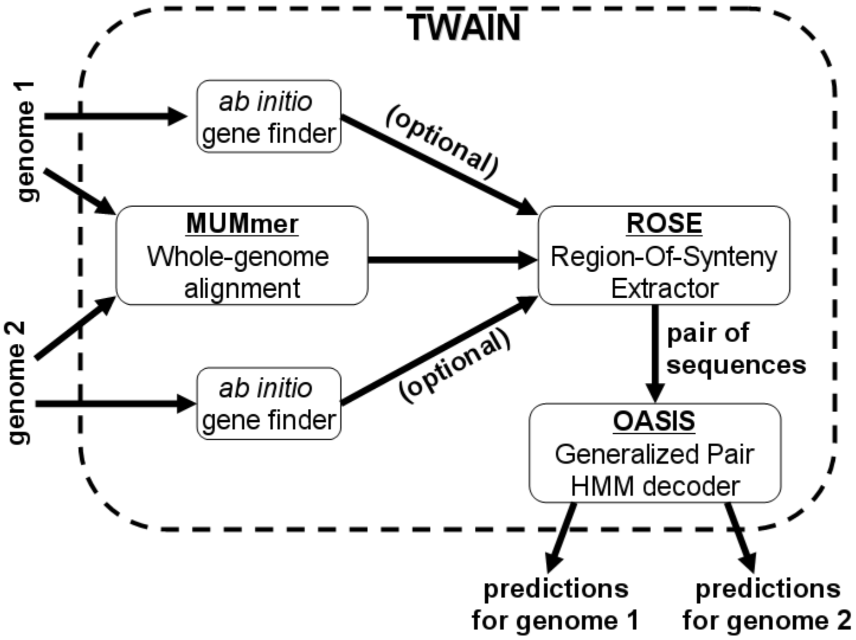

Fig. 1. Software architecture for a Generalized Pair Hidden Markov Model gene finder, TWAIN. Two genomes are read as inputs, and predictions for both genomes are emitted. The three major components of the system are a whole-genome alignment component (MUMmer), a region-of-synteny extractor (ROSE), and a GPHMM decoder (OASIS). Optional *ab initio* gene predictions may also be used by ROSE to more effectively cluster syntenic regions, but these are not obligatory.

Because PROmer can report hits for any of the four strand combinations of the two contigs and because TWAIN's GPHMM predicts genes only on the forward strand, ROSE employs a standard *longest common subsequence* algorithm (Cormen *et al.*, 1990) to find the longest series of PROmer hits having consistent orientations, and then performs reverse-complementation as necessary to render the putative exons underlying those PROmer hits onto the forward strand in both genomes.

In addition, when *ab initio* predictions are available as additional inputs, ROSE attempts to cluster PROmer hits in such a way as to avoid interrupting a likely gene in either genome. When a predicted gene overlaps an HSP, we use the prediction to extend the boundaries of the putative conserved coding region. Because the GPHMM can predict exons only in places where a PROmer hit was found, ROSE invokes PROmer a second time with a more liberal parameterization on the regions between the individual HSPs originally identified, to find evidence of less strongly conserved exons. For this second set of HSPs, no LIS-like filtering is performed except to ensure that all HSPs within a cluster are on the same strand. In this way, we try to attain high sensitivity to avoid over-constraining the GPHMM. The sequence is segmented around the resulting clusters, with a default margin of 1000 bp being added on either side before the pair of sequence segments are provided to OASIS.

Finally, NUCmer's HSPs are combined into a single, global, "guide" alignment by first applying an LIS-like algorithm to the NUCmer HSPs (via NUCmer's *–g* option) and then employing the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) as necessary to align the gaps between HSPs. This step is necessary in order to allow OASIS to utilize the global nucleotide guide alignment for evaluating conservation in arbitrarily long noncoding regions. Similarly, each PROmer alignment is extended with diagonal cells in both directions until either an in-frame stop codon is encountered or the beginning or end of the sequence is reached. The separate PROmer HSPs are not combined as with NUCmer. All alignment parameters can be changed by the user without recompiling the source code.

A more detailed description of the algorithms used by ROSE can be found on the web page cited in the abstract.

OASIS: Optimal Alignment of Structures In Synteny

The structure of the GPHMM used by TWAIN is depicted in Fig. 2. The GPHMM decoder in TWAIN is encapsulated in the C++ program OASIS. OASIS utilizes a novel modification to the Viterbi dynamic programming algorithm to find the optimal path ϕ according to Equations (4,5). Because each state in ϕ corresponds to a pair of features, the OASIS algorithm can be conceptualized as an alignment algorithm between gene parses, as illustrated in Fig. 3.

To formalize the notion of aligning gene parses, define a very minimal set of *signal types* $\sigma=\{\texttt{ATG},\texttt{TAG},\texttt{GT},\texttt{AG}\}$ denoting start codon, stop codon, donor site, and acceptor site, respectively, of any appropriate consensus (e.g., $\texttt{TAG}\equiv\{\text{TAG, TGA, TAA}\}$ typically). The generalization of the methods described below to incorporate promoters, poly-adenylation signals, and other signal types will be obvious. Denote a *parse graph G* by (V,E) for vertex set $V\subset\sigma\times\mathbb{N}$ and directed edge set $E\subset V\times V$, so that each vertex $v=(s,x)$ corresponds to a putative signal of type $s\in\sigma$ at position x in one of the two genomes, and each directed edge corresponds to a putative exon, intron, or intergenic region. In particular, the following edge types are represented: $\texttt{ATG}\rightarrow\texttt{TAG}$, $\texttt{ATG}\rightarrow\texttt{GT}$, $\texttt{GT}\rightarrow\texttt{AG}$, $\texttt{AG}\rightarrow\texttt{GT}$, $\texttt{AG}\rightarrow\texttt{TAG}$, and $\texttt{TAG}\rightarrow\texttt{ATG}$. Two hypothetical parse graphs are illustrated in Fig. 3 (one along each axis of the matrix).

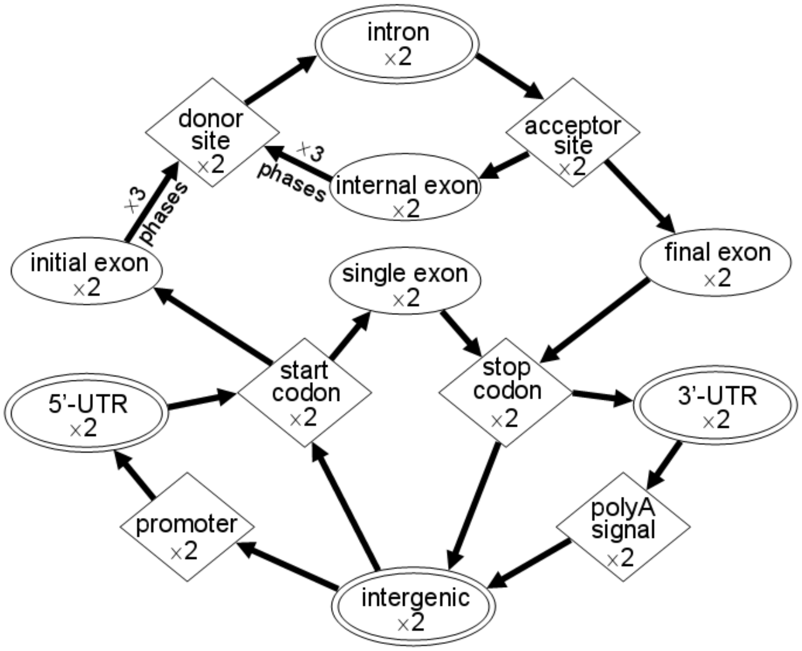

Fig. 2. State-transition diagram for the GPHMM used by TWAIN. States corresponding to fixed-length signals are shown as diamonds and those corresponding to variable-length features (e.g., exons and introns) are shown as ovals. Each state emits pairs of features, as indicated by the "×2" in each state. Arrows denote legal transitions. The silent initial and final states are omitted for clarity. States with double borders have implicit edges from the initial state and to the final state; starting or ending in an intron constitutes generation of a partial gene.

The GHMM gene finder TIGRscan (Majoros *et al.*, 2004) is used as a subroutine by OASIS to construct a parse graph for each of the two genomes. TIGRscan constructs its graph left-to-right by sliding its signal sensors (typically WMM, WAM, or MDD (Burge and Karlin, 1997) models) along the sequence and allocating a vertex at each high-scoring position. Nonconsensus splice sites are permitted. Each new vertex is attached via an edge back to any previous vertex of the appropriate type (unless eclipsed in all frames by stop codons). In the case of noncoding predecessor edges, only the top-scoring *r* predecessor edges are kept at each vertex, for some user-defined *r*; for coding edges, no such limit is imposed. Vertices unreachable from either end of the graph are discarded. This heuristic tends to retain only the high-scoring subgraphs, and was found to be invaluable for maintaining speed and space efficiency (see Results). Each edge is annotated by TIGRscan with three scores associated with the corresponding feature in the three possible phases (or a single phase for noncoding edges). These scores comprise the $P_e(S_{i,1}|q_i,d_{i,1})$ $P_t(q_i|q_{i-1})$ $P_d(d_{i,1}|q_i)$ term corresponding to Equations (4,5), and are computed as described in (Majoros *et al.*, 2005).

The vertices in these two graphs are then placed along the axes of a two-dimensional dynamic programming matrix as shown in Fig. 3. Because this matrix can become very large in practice, OASIS employs a sparse representation of the matrix by eliminating all but the most promising regions from consideration.

In particular, each PROmer alignment is mapped into the signal space of the OASIS matrix (using the genome coordinates of the putative signals in the two parse graphs as a frame of reference). Only those OASIS cells within a user-specified nucleotide distance Δ of a PROmer alignment path are retained, and only those for which the two corresponding signals are of the same signal type. The result is typically a rather sparse matrix (see Results).

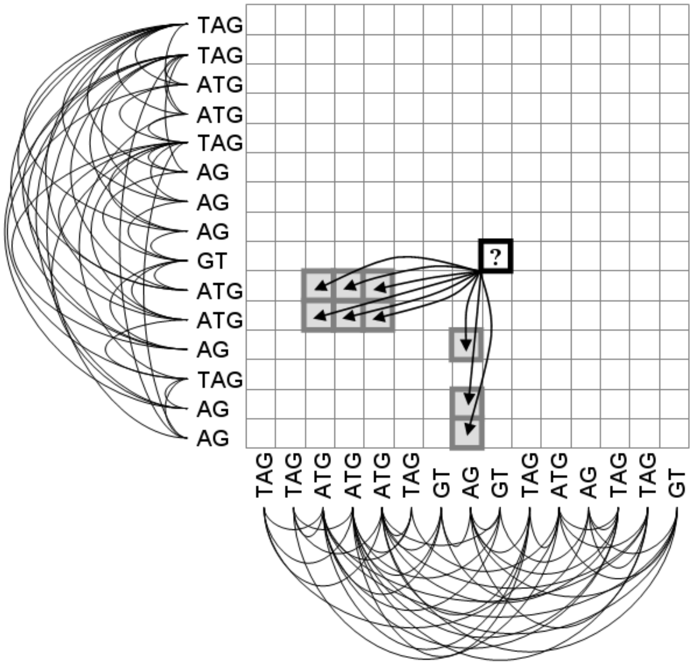

Fig. 3. A conceptual representation of the full OASIS dynamic programming matrix. A *parse graph* for each of the two genomes is shown along each axis. Rows and columns of the matrix correspond to vertices in the two parse graphs. The vertices in the parse graphs correspond to signals such as splice sites and start and stop codons, and edges correspond to features such as exons, introns, and intergenic regions. Edges are implicitly directed 5'-to-3' along the sequence (arrows not shown for clarity). Interior to the matrix is shown a single step of the *trellis* linking process, in which the cell marked '?' is linked back to the optimal predecessor cell in each pair of phases. Each edge in the trellis corresponds to a pair of edges from the two parse graphs.

Each OASIS cell is then attributed with a set of predecessor links (Fig. 3) formed by taking the cartesian product of the outgoing edges on the two vertices corresponding to that cell. In this way, each OASIS link has associated with it two TIGRscan edges, one from each parse graph for the two genomes. The resulting set of OASIS cells and their predecessor links form a *trellis*. It should be evident that a path through the trellis outlines an isomorphism between subgraphs of the two parse graphs, and that these subgraphs outline corresponding gene predictions (although ensuring that those gene predictions are well formed requires tracking of phase constraints using the usual *mod 3* arithmetic employed in GHMM gene finders, as described in (Majoros *et al.*, 2005)).

Evaluation of the matrix is then performed according to Equations (4,5). Each coding link in the trellis is annotated with nine scores (or one for noncoding links), corresponding to the cartesian product of the three possible phases of the associated features in the two genomes. Storing all nine scores separately is necessary for coding edges in order to avoid greedy behavior in the GPHMM decoding algorithm. The dynamic programming recurrence for this algorithm is as follows:

$$\beta(i,j) = \mathbf{max}_{(h,k)\in pred(i,j)} \Big[\beta(h,k)\, P_e(S_{h:i,1}|q_{h:i},d_{h:i})\, P_e(S_{i,1}|q_i) \qquad (6)$$
$$P_{cond}(S_{k:j,2}|S_{h:i,1},q_{h:i},d_{h:i})\, P_d(d_{h:i}|q_{h:i})\, P_t(q_i|q_h)\Big]$$

where (i,j) and (h,k) address cells in the OASIS matrix, pred(i,j) is the set of links to predecessor cells in the trellis, $S_{i,1}$ is the fixed-length sequence in the immediate vicinity of signal i in the parse graph for genome 1, $S_{h:i,g}$ is the variable-length sequence between signals h and i in genome $g\in\{0,1\}$, q_i is the state corresponding to signal i, and $q_{h:i}$ is the variable-duration state corresponding to $S_{h:i,g}$. The product $P_e(S_{h:i,1}|q_{h:i},d_{h:i})\, P_e(S_{i,1}|q_i)\, P_d(d_{h:i}|q_{h:i})\, P_t(q_i|q_h)$ is cached in each edge of parse graph 1 so that these terms need not be re-evaluated for each cell in the OASIS matrix; these terms are evaluated by TIGRscan. $\beta(h,k)$ is the inductive score which is stored separately for each of the nine phase pairs in each cell of the OASIS matrix. Note that the phases of putative orthologous exons are not required to match; phase is tracked only to enforce phase constraints separately in the two genomes, as in a typical GHMM gene finder. The $P_{cond}(S_{k:j,2}|S_{h:i,1},q_{h:i},d_{h:i})$ term is approximated as:

$$\left| P_{match} \right|^{P_{ident} d_{h:i}} \left| 1 - P_{match} \right|^{(1-P_{ident})d_{h:i}} \qquad (7)$$

where P_{ident} (percent identity or similarity) is estimated using an approximate alignment procedure, as described below. P_{match} is a parameter to the GPHMM which is estimated during training to reflect the probability of a single residue match in a coding or noncoding alignment (as per the coding/noncoding status of state $q_{h:i}$). A pointer to the predecessor (h,k) selected by the **max** term above is stored in one of the nine slots at each cell (according to phase). Once the matrix has been evaluated the highest-scoring path through the trellis is found using a standard trace-back procedure with the usual phase constraints for coding regions.

The P_{ident} term is estimated using an approximate alignment procedure (Fig. 4a), as follows. In each cell c of a PROmer or NUCmer alignment is stored two values: μ_c, the cumulative number of matches from the beginning of the alignment to the current cell c; and λ_c, the length of the alignment up to cell c. For PROmer alignments a BLOSUM score greater than zero is counted as a match. Which BLOSUM matrix to use is specified in a configuration file loaded at run-time. Because a pair of OASIS cells A and B may not fall directly on the PROmer or NUCmer alignment, additional indel terms δ_A and δ_B are assessed by finding the nearest alignment cells z and y to OASIS cells A and B, respectively, using a binary search, and then simply assessing the distance from each OASIS cell to the corresponding alignment cell in nucleotide space. In this way, P_{ident} can be estimated using:

$$P_{ident} = (\mu_z\text{-}\mu_y)/(\lambda_z\text{-}\lambda_y+\delta_A+\delta_B), \qquad (8)$$

where the putative feature pair extends from OASIS cell B to A. When no PROmer or NUCmer alignment is near enough to A and B to give a nonnegative value for the numerator of Equation (8), we set P_{ident}=0.

Fig. 4b shows that these approximate alignment scores (x-axis) correlate fairly well with full Needleman-Wunsch alignment scores (y-axis), though the approximate scores appear to underestimate the degree of conservation for the less-conserved features, and there is a non-negligible amount of variance. Percent identity was used for both alignment scores in the graph. The sequences which were aligned were selected randomly from *Aspergillus fumigatus* and *A. nidulans* ORFs and filtered so as to consider only those which differed by no more than 5% in length and which overlapped a PROmer HSP. Only nucleotide sequences were aligned.

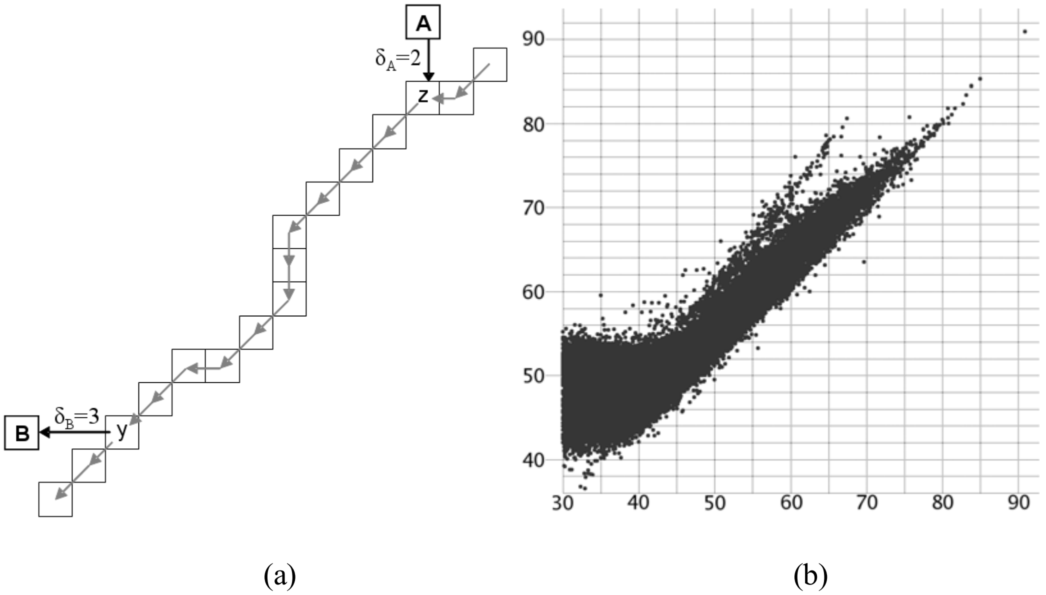

Fig. 4. (a) Approximation of percent identity or percent similarity (P_{ident}) between a pair of features delimited by cells A and B in the OASIS matrix. An approximate alignment is rapidly formed by jumping from OASIS cell A to the nearest "guide" alignment cell z, then following along the precomputed alignment from z to y, and then jumping from y to OASIS cell B. Jumps to and from the guide alignment incur indel penalties δ_A and δ_B. The remaining portion of the alignment is scored in constant time by simple subtraction. (b) Correlation between approximate alignment scores (x) and full Needleman-Wunsch alignment scores (y), using percent identity as the alignment score in both cases. Correlation coefficient was 0.92 for points above (0.5,0.5).

Results

We compared the performance of TWAIN to the non-comparative gene finder TIGRscan on a set of 147 *A. fumigatus* × *A. nidulans* likely orthologues. The orthologues were identified by using BLAST (Altschul *et al.*, 1990) to obtain the mutual best-match triples between *A. fumigatus*, *A. nidulans*, and *A. oryzae* auto-annotated genes (Nierman *et al.*), as in the well-known COG method (Tatusov *et al.*, 2000; Tatusov *et al.*, 2003). The auto-annotation pipeline used for this genome did not at this time include TWAIN or any of its components. The *A. fumigatus* and *A. nidulans* pairs were filtered to eliminate those having unequal numbers of exons, those not aligned by BLAST over their full length, and those not having an amino acid similarity of 90% or higher. This filtering was necessary to obtain a relatively high confidence test set, since no set of confirmed orthologues was available at the beginning of our study. Test genes were on average 1840 ± (SD) 1193 bp long (range: 655-7451) and consisted of 3.4 ± 1.6 coding exons (range: 1-8). The underlying GHMM was trained on a set of 9796 *A. nidulans* and 9368 *A. fumigatus* annotated genes. BLAST was used to ensure that the training and test sets were disjoint.

Results are shown in Table 1. As can be seen from the table, OASIS produces higher accuracy than TIGRscan even though TIGRscan already performs very well on this test set. Despite the stringent filtering of the test set we feel these results illustrate well the value of employing homology in gene finding. Though this elevated performance of the syntenic over the non-syntenic gene finder can be expected only for those gene pairs exhibiting a suitable degree of homology, for certain pairs of organisms we expect these gains in accuracy to make a noticeable improvement in the quality of annotations in the more highly conserved portions of those genomes.

	nucleotide			**splice sites**		**start/stop codons**		**exons**			**genes**	
	Sn	*Sp*	*F*	*Sn*	*Sp*	*Sn*	*Sp*	*Sn*	*Sp*	*F*	*Sn*	#
TIGRscan	99%	100%	99%	89%	81%	81%	80%	78%	73%	75%	54%	79
TWAIN	99%	100%	99%	94%	88%	92%	92%	89%	85%	87%	74%	109

Table 1: Accuracy results for OASIS applied to *Aspergillus fumigatus* using *A. nidulans* as the reference genome, and TIGRscan applied to *A. fumigatus*. *Sn*=TP/(TP+FN), *Sp*=TP/(TP+FP), *TP*=true positives, *FP*=false positives, *TN*=true negatives, *FN*=false negatives. *F*=2SnSp/(Sn+Sp). For nucleotides a positive is a coding nucleotide. For exons a true positive had both begin and end coordinates exactly correct. For genes a true positive had all exons correct. Numbers of genes predicted corrected are shown in the column marked '#'.

OASIS was able to process the 147-gene set in just under one hour on a laptop computer equipped with a 1.6 GHz Intel Centrino 725 processor and 512 Mb of RAM. Most runs required under 50 Mb of RAM. ROSE (and all subprocesses, including MUMmer) was able to process both genomes in under 30 minutes on a 2.4 GHz Intel dual-Xeon machine and consumed 689 Mb of RAM. Sequences provided by ROSE to OASIS were on average 3830 ± 1190 bp in length (range: 2652-9448). The OASIS dynamic programming matrix remained quite sparse during all runs, with on average only 4% ± 1.7% of cells in the matrix being allocated (range: 1%-8%).

We found through trial and error the optimal P_{match} (probability of the GPHMM emitting a pair of matching residues) values for coding and noncoding features for our test set to be 0.64 and 0.58, respectively. These differ from the observed mean similarity of 0.80 and mean identity of 0.47 for coding and noncoding alignments, respectively, in the test set. As a possible explanation for this discrepancy, we draw attention to the fact that while 0.80 and 0.47 would seem to be the maximum likelihood estimates for these two parameters based only on the alignments in the test set, they do not necessarily represent the ML estimates for the globally optimal parameterization according to Equation (3). A global optimization procedure would have to take into account the interactions between the different terms in the formula for $P(\phi|S)$, the probability of a parse ϕ given the sequence S (Majoros and Salzberg, 2004).

That such an interaction exists involving P_{cond}, and is nontrivial, is suggested by our informal observation that for P_{match} values of 0.80 (coding) and 0.47 (noncoding) the gene finder tended to predict additional exons of small size (~17bp) which were not present in the annotation. Though a small number of these extra exons may be true exons which were missing from the annotations in the test set, we rather suspect the P_{match} values were largely to blame by causing prediction of additional small exons exhibiting spurious patterns of amino acid conservation. Thus, in order to prevent the alignment term from dominating the optimization step during gene prediction to the detriment of overall accuracy, it may be necessary in many cases to modify the P_{match} values so as to strike a more suitable balance between the homology and *ab initio* forms of evidence. Alternatively, some statistical test may be necessary to reject evidence of conservation based on small sample sizes. A theoretically sound way of incorporating such a test might involve adding states to OASIS to represent *paired nonconserved* features in addition to the *paired conserved* states already in the GPHMM; then the results of an appropriate test on feature length might influence the probability of transitioning into the conserved *vs.* nonconserved state.

Discussion

Although the results reported above are very encouraging, there are a number of possibilities for further improvement which we would like to see investigated. Chief among these is the ability to correctly predict orthologues with inserted introns and/or exons, which to our knowledge is not possible with the current generation of GPHMM implementations. Currently, exons must be emitted in pairs, as must introns. Proper handling of unequal numbers of exons in OASIS would require changes both to the strict signal-type matching discipline in the OASIS matrix (thereby allowing cells to correspond to signals of differing types) as well as a means for detecting which additional regions of the sparse matrix must be allocated in order to accommodate exons not detected by a PROmer HSP. Unless the presence of nonconserved exons can be reliably detected at matrix allocation time (i.e., without first evaluating the full matrix), such changes to OASIS are likely to expand the allocated portion of the dynamic programming matrix significantly, thereby increasing execution time and memory requirements, perhaps beyond what is practical. We are currently considering possibilities for doing this efficiently. Although the prediction of nonconserved exon structures is apparently easier in the case of non-generalized pair HMMs (e.g., Meyer and Durbin, 2002), we would like to be able to do so with a generalized pair HMM so as to benefit from all of the advantages of model generalization as documented in the GHMM literature (e.g., Kulp *et al.*, 1996).

Of the other heuristic elements of TWAIN, those which seem to us the most likely targets for profitable improvement are those associated with the use of approximate alignments. The effective use of approximate alignments to improve the run-time efficiency of a GPHMM gene finder was first demonstrated in SLAM (Alexandersson *et al.*, 2003), although a number of non-GPHMM comparative gene finders have used precomputed alignments as well, including GLASS/ROSETTA (Batzoglou *et al.*, 2000), SGP1 (Wiehe *et al.*, 2001), EvoGene (Pedersen and Hein, 2003), and (to some degree) DoubleScan (Meyer and Durbin, 2002). Because current commodity computing systems cannot compute full pairwise alignments between all cells in the GPHMM dynamic programming matrix with acceptable rapidity, the use of approximate alignments in GPHMM gene finders is likely to remain necessary in some form for the foreseeable future, at least on common computing hardware. There are, however, a number of ways in which the approximate alignment procedure described herein might be extended to allow for possibly more accurate estimation of alignment scores. These include the use of larger numbers of pre-computed "guide" alignments, both in nucleotide and amino acid space, for each putative orthologue pair; the use of alternate alignment scoring functions; the use of a more accurate paired duration probability function, $\psi_d(d_{i,1},d_{i,2}|q_i)$; and the use of a more principled model of nucleotide and amino acid conservation based on evolutionary distances and estimates of substitution rates, perhaps similar to methods employed in Evolutionary HMMs (e.g., Pedersen and Hein, 2003) or Phylogenetic HMMs (e.g., Siepel and Haussler, 2004).

It has been debated previously (e.g., Batzoglou *et al.*, 2000; Wiehe *et al.*, 2001; Zhang *et al.*, 2003) whether one might expect a rather limited range of evolutionary distances in which two genomes must fall in order for comparative gene-finding methods to provide an advantage over *ab initio* methods. Though we cannot here resolve this debate, comparison of the nucleotide and amino acid alignment scores for known exons and introns for a number of the genes in our test set show that very often the amino acid conservation was higher than the nucleotide conservation for the exons (91%), and vice-versa for the introns (99.7%). Conventional wisdom suggests that the difference between nucleotide and amino acid conservation for coding and non-coding features of orthologues should influence the accuracy gains associated with the use of homology evidence during gene prediction, yet to our knowledge reliable bounds have not been placed on the ideal divergence by empirical studies, though some simulations have been performed (e.g., Zhang *et al.*, 2003). We hope that the availability of our open-source comparative gene finder will allow this question to be more directly addressed in the near future through large-scale comparative studies, as larger numbers of closely related genomes become available.

In summary, although TWAIN appears to perform very well on this pair of fungal genomes, we believe additional gains in accuracy are likely possible. Our immediate goals are to explore possible enhancements to our implementation after more fully characterizing its current strengths and weaknesses on a larger data set. We are now in the process of organizing a larger-scale experiment to study its accuracy on a wider array of organisms and under a wider range of parameterizations. We hope to report the results of those experiments at a future date, and in the meantime we encourage others to consider using our software for annotation and/or computational research purposes, in hopes of improving the state of the art in comparative gene finding.

References

Alexandersson,M., Cawley,S. and Pachter,L. (2003) SLAM: Cross-species gene finding and alignment with a generalized pair hidden Markov model. *Genome Res.*, **13**, 496-502.

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

Batzoglou,S., Pachter,L., Mesirov,J.P., Berger,B. and Lander,E.S. (2000) Human and mouse gene structure: comparative analysis and application to exon prediction. *Genome Res.,* **10**, 950-958.

Burge,C. (1997) Identification of genes in human genomic DNA. PhD thesis, Stanford University, Stanford, CA.

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

Cawley,S.E., Wirth,A.I. and Speed,T.P. (2001) Phat—a gene finding program for *Plasmodium falciparum. Mol. & Biochem. Parasitology*, **118**, 167-174.

Cormen,T.H., Leiserson,C.E. and Rivest,R.L. (1990) *Introduction to Algorithms*. MIT Press: Cambridge.

Kellis,M., Patterson,N., Endrizzi,M., Birren,B. and Lander, E.S (2003) Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, **423**, 241-254.

Korf,I. (2004) Gene finding in novel genomes. *BMC Bioinf.*, **5**, 59.

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 *Proceedings of 4th International Conference on Intelligent Systems for Molecular Biology (ISMB'96)*, pp. 134-142.

Kurtz,S., Phillippy,A., Delcher,A.L., Smoot,M., Shumway,M., Antonescu,C. and Salzberg,S.L. (2004) Versatile and open software for comparing large genomes. *Genome Biol.*, **5**, R12.

Majoros,W.M., Pertea,M., Delcher A.L. and Salzberg,S.L. (2005) Efficient decoding algorithms for generalized hidden Markov model gene finders. *BMC Bioinformatics*. Accepted for publication.

Majoros,W.M., Pertea,M. and Salzberg,S.L. (2004) TIGRscan and GlimmerHMM: two open-source *ab initio* eukaryotic gene finders. *Bioinformatics*, **20**, 2878-2879.

Majoros,W.M. and Salzberg,S.L. (2004) An empirical analysis of training protocols for probabilistic gene finders. *BMC Bioinformatics* **5**, 206.

Meyer,I.M. and Durbin,R. (2002) Comparative *ab initio* prediction of gene structures using pair HMMs. *Bioinformatics,* **18**, 1309-1318.

Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.,* **48**, 443-453.

Nierman,W.C. *et al.*, Genomic sequence of the pathogenic and filamentous fungus, *Aspergillus fumigatus*. Submitted.

Novichkov,P.S., Gelfand,M.S. and Mironov,A.A. (2001) Gene recognition in eukaryotic DNA by comparison of genomic sequences. *Bioinformatics* **17**, 1011-1018.

Pachter,L., Alexandersson,M. and Cawley,S. (2002) Applications of generalized pair hidden Markov models to alignment and gene finding problems. *Journal of Comp. Biol.* **9**, 389-399.

Pedersen,J.S. and Hein,J. (2003) Gene finding with a hidden Markov model of genome structure and evolution. *Bioinformatics,* **19**, 219-227.

Rabiner,L.R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, **77**, 257-285.

Siepel,A. and Haussler,D. (2004) Computational identification of evolutionarily conserved exons. *RECOMB'04*, March 27-31, 2004, San Diego, CA.

Stanke,M. and Waack,S. (2003) Gene prediction with a hidden Markov model and a new intron submodel. *Bioinformatics*, **19**, II215-II225.

Tatusov,R.L., Fedorova,N.D., Jackson,J.D., Jacobs,A.R., Kiryutin,B., Koonin,E.V., Krylov,D.M., Mazumder,R., Mekhedov,S.L., Nikolskaya,A.N., Rao,B.S., Smirnov,S., Sverdlov,A.V., Vasudevan,S., Wolf,Y.I., Yin,J.J. and Natale,D.A.. (2003) The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, **4**, 41.

Tatusov,R.L., Galperin,M.Y., Natale,D.A. and Koonin,E.V. (2000) The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Res.*, **28**, 33-36.

Wiehe,T., Gebauer-Jung,S., Mitchell-Olds,T. and Guigó,R. (2001) SGP-1: prediction and validation of homologous genes based on sequence alignments. *Genome Res.,* **11**, 1574-1583.

Zhang,L., Pavlovic,V., Cantor,C.R. and Kasif,S. (2003) Human-mouse gene identification by comparative evidence integration and evolutionary analysis. *Genome Res.,* **13**,1190-1202.

Acknowledgements

This work was supported in part by NIH grant R01-LM007938. ROSE and OASIS were developed by MP and WHM, respectively, under the supervision of SLS. We thank Jennifer Wortman, Jonathan Crabtree, Jay Sundaram, and Christopher Hauser for providing the training and test data for this study. WHM thanks Ian Korf and Mark Yandell for useful discussions and advice.