

SLAM – Cross-species Gene Finding and Alignment with a Generalized Pair Hidden Markov Model

Marina Alexandersson
Department of Statistics,
U.C. Berkeley

marina@stat.berkeley.edu

Simon Cawley
Affymetrix Inc.,
Santa Clara, CA

simon_cawley@affymetrix.com

Lior Pachter
Department of Mathematics,
U.C. Berkeley

lpachter@math.berkeley.edu

ABSTRACT

Comparative based gene recognition is driven by the principle that conserved regions between related organisms are more likely than divergent regions to be coding. We describe a probabilistic framework for gene structure and alignment that can be used to simultaneously find both the gene structure and alignment of two syntenic genomic regions. A key feature of the method is the ability to enhance gene predictions by finding the best alignment between two syntenic sequences, while at the same time finding biologically meaningful alignments that preserve the correspondence between coding exons. Our probabilistic framework is the generalized pair hidden Markov model, a hybrid of generalized hidden Markov models which have been used previously for gene finding, and pair hidden Markov models which have applications to sequence alignment. We have built a gene finding and alignment program called **SLAM** which aligns and identifies complete exon/intron structures of genes in two related but unannotated sequences of DNA. **SLAM** is able to reliably predict gene structures for any suitably related pair of organisms, most notably with fewer false positive predictions than previous methods (examples are provided for *H. sapiens*/*M. musculus* and *P. falciparum*/*vivax* comparisons). Accuracy is obtained by distinguishing conserved non-coding sequence (CNS) from conserved coding sequence. CNS annotation is a novel feature of **SLAM** and may be useful for the annotation of UTRs, regulatory elements and other non-coding features.

1. INTRODUCTION

The idea of comparing organisms in order to further the understanding of their biology has been a central theme in biology, arguably originating with the work of Darwin who formalized the theory of evolution by observing the similarities and differences between related organisms. The comparative method itself has evolved, advancing much in recent years with the ability to compare the genomic sequences of organisms. These comparisons have yielded many new

results, often leading directly to important biological discoveries (for a recent example see [1]). Indeed, motivated by the success of comparative genomics in identifying regulatory elements, Hardison et al. [2] suggested sequencing the mouse genome for the purpose of annotating the human genome.

The observation that human/mouse genomic comparisons are able to highlight coding exons led to the realization by Lander [3] that the comparative sequencing approach to biological discovery should be formalized and automated. This resulted in an initiative which produced the **ROSETTA** program [4], the first automated program for annotating human genes using syntenic unannotated mouse genomic DNA. The comparative approach was subsequently adopted by several groups, resulting in the **CEM** program [5], **TWINSCAN** [6], **SGP-1** [7] and **SGP-2** (paper in preparation, [8]). These programs differ from the homology based gene finders such as **PROCRUSTES** [9], **GENOMESCAN** [10] and **GENEWISE2** [11], in that rather than using protein homologs or confirming EST evidence to help in coding exon prediction, they compare two identical types of sequences (genomic DNA). This distinction is subtle but important; comparative based gene finders can use extra information such as gene structure and splice site conservation, introducing complications different to the issues arising in other homology based approaches.

At the same time as gene finding was moving toward the comparative approach, a similar development was taking place in the alignment community. Alignment programs such as **BLAST** [12] had traditionally been based on pure sequence comparison. Both in **BLAST** as well as in other methods there had been no attempt to incorporate the annotation of the sequences being aligned into the alignment program. Because biological sequences do not display random patterns of conservation, the consideration of biological features during alignment can greatly improve performance. An excellent example of this is **WABA** (Wobble Aware Bulk Aligner) [13] which takes advantage of the 3rd base wobble in coding exons to improve alignment and was successfully applied towards the problem of aligning the *C. elegans* and *C. briggsae* genomes.

In this paper we describe a program that places the annotation and alignment problems on an equal footing. Our probabilistic model is a *generalized pair hidden Markov model* (GPHMM). Generalized hidden Markov models (GHMMs)

have been applied successfully in gene finding programs such as **GENSCAN** [14] or **GENIE** [15]. Pair hidden Markov models (PHMMs) have been used for alignment, and can be shown to be equivalent to the Needleman-Wunsch [16] alignment method [17, 18]. The GPHMM we have developed directly generalizes both of these types of HMMs. As a special case, by appropriately altering model parameters, our method can be made equivalent to GHMM-based single organism gene finders like **GENSCAN** or **GENIE**, or to comparative gene finders such as **ROSETTA** (which separates the steps of alignment and gene finding). We have built a program called **SLAM** which implements these ideas, and can be used to annotate syntenic sequences by finding coding exons and conserved non-coding sequences, or as a global alignment program which takes advantage of the biological features of the sequences to improve the accuracy of the alignments.

2. RESULTS

The **SLAM** program was tested on the **ROSETTA** test set [4] of 117 single-gene sequences as well as on the multi-gene HoxA cluster ([19], 220 Kb) and the Elastin gene region (accession numbers NT_025776 and NT_014920, 390 Kb). **SLAM** was compared to the following programs:

- **GENSCAN** [14], makes gene predictions in genomic DNA from a single organism.
- **ROSETTA** [4], uses syntenic DNA pairs to make gene predictions in one sequence.
- **SGP-1** [7], uses syntenic DNA pairs to make gene predictions in *both* sequences.
- **SGP-2** (paper in preparation [8]), predicts genes in one sequence incorporating as evidence matches to a collection of informant sequences.
- **TWINSKAN** [6], predicts genes in one sequence incorporating as evidence matches to a collection of informant sequences.

Note that the **TWINSKAN** web-server allows for the specification of a custom informant sequence to be used instead of the default informant sequence database. By supplying the syntenic mouse DNA as a custom informant sequence for a human region it is therefore possible to run **TWINSKAN** on syntenic DNA pairs — we use the modifier **TWINSKAN.p** to label runs of this type. The most direct comparison is then between **ROSETTA**, **SGP-1**, **SLAM** and **TWINSKAN.p**, since all run on a syntenic pair of genomic DNA sequences. **TWINSKAN** and **SGP-2** fall into their own category, each incorporating matches against a database of mouse sequences to predict genes in human, and **GENSCAN** serves as a benchmark, making gene predictions using only one sequence.

Results for **GENSCAN** and **TWINSKAN** were obtained by submitting the test sets to their servers (<http://genes.mit.edu/GENSCAN.html> and <http://genes.cs.wustl.edu/> respectively), the results of **SGP-1** on the **ROSETTA** set were retrieved from [7]. **SGP-2** results for HoxA and Elastin were obtained from Guigó [8]. The programs were compared using standard performance measures [20, 14]. The results of the programs on the test

sets are summarized in Table 1. The table presents the sensitivity (SN) and the specificity (SP) both at the nucleotide and exon level, the approximate correlation (AC), as well as rates for missed (ME) and wrong (WE) exons (false positives not overlapping any true exons).

Perhaps the most striking aspect of the results in Table 1 is the difference in performance between the class of programs operating on syntenic pairs (**ROSETTA**, **SGP-1**, **SLAM** and **TWINSKAN.p**) and the class of programs operating on human sequence using matches against a mouse database (**SGP-2**, **TWINSKAN**). The result is not unexpected — if homology against a large database of sequences is used to boost exon scores, this will naturally include more false positive alignments, leading to a degradation in sensitivity (the difference is particularly large in the case of HoxA, where the sensitivity achieved is even lower than that of a single-organism gene finder). At the same time, the increase in sensitivity when using homology against a large database is negligible. It is clear that whenever possible, it is better to operate on a syntenic DNA pair (though of course in practice the finished genomic data may not be available).

Analysis of the programs operating on syntenic pairs on the **ROSETTA** test set shows that while **SLAM** nucleotide sensitivity is slightly lower than for **TWINSKAN.p**, the specificity is significantly higher with half as many nucleotides being incorrectly predicted as coding. At the exon level **ROSETTA** and **TWINSKAN.p** perform better. **SLAM**'s high nucleotide scores in conjunction with the low wrong and missed exon rates suggest that it is getting exon boundaries slightly wrong rather than missing them entirely. The current model used for a human-mouse splice site pair treats the splice site sequences as independent in each organism. Clearly, modeling the significant conservation in splice site pairs will improve exon-level performance. Examination of the longer HoxA and Elastin regions shows that **SLAM**'s specificity is consistently higher. The lower sensitivity rates for **SLAM** on these regions is due in part to inaccurate approximate alignments (a pre-processing step done with **AVID** [21] to reduce the computational complexity of the GPHMM); this problem which arises with longer (more difficult to align) regions should be fixed with the forthcoming implementation of more sophisticated approximate alignment methods.

There are a number of reasons why we believe **SLAM** should be highly specific. A notable property shared by **SLAM** and **SGP-1** is that the gene prediction is performed symmetrically in both sequences. In addition to requiring good alignment between exons, this has the effect of requiring conservation of exon-order and frame consistency in both sequences. Another important and novel feature of **SLAM** is the prediction of conserved non-coding sequence (CNS). The annotation of CNSs allows for the distinction between conserved coding and conserved non-coding sequence in a probabilistic manner.

It has been observed [4, 22] that in the case of human/mouse comparisons there is a lot of non-coding conservation to be found, including UTR, regulatory element and other biologically related conservation, and also non-functional background conservation. The CNS state significantly lowers the false positive rate by eliminating the consideration of

Test set	Nucleotide level			Exon level				
	SN	SP	AC	SN	SP	(SN+SP)/2	ME	WE
The ROSETTA set								
ROSETTA	0.935	0.978	0.949	0.833	0.829	0.831	0.048	0.047
SGP-1	0.940	0.960	0.940	0.700	0.760	0.730	0.120	0.040
SLAM	0.951	0.981	0.960	0.783	0.755	0.769	0.038	0.057
TWINSKAN.p	0.960	0.941	0.940	0.855	0.824	0.840	0.045	0.081
TWINSKAN	0.984	0.889	0.923	0.839	0.767	0.803	0.034	0.118
GENSCAN	0.975	0.908	0.929	0.817	0.770	0.793	0.057	0.107
HoxA								
SLAM	0.852	0.896	0.864	0.727	0.533	0.630	0.000	0.333
TWINSKAN.p	0.976	0.829	0.896	0.773	0.531	0.652	0.000	0.312
TWINSKAN	0.949	0.511	0.704	0.591	0.173	0.382	0.000	0.707
SGP-2	0.640	0.637	0.619	0.409	0.173	0.291	0.091	0.596
GENSCAN	0.932	0.687	0.796	0.545	0.235	0.390	0.000	0.569
Elastin								
SLAM	0.876	0.981	0.926	0.802	0.859	0.831	0.121	0.059
TWINSKAN.p	0.942	0.950	0.945	0.879	0.889	0.884	0.066	0.056
TWINSKAN	0.933	0.877	0.903	0.835	0.826	0.831	0.110	0.120
SGP-2	0.755	0.998	0.873	0.593	0.900	0.291	0.352	0.017
GENSCAN	0.947	0.766	0.852	0.835	0.731	0.783	0.121	0.231

Table 1: Results on the test sets. The measures of sensitivity $SN = \frac{TP}{TP+FN}$ and specificity $SP = \frac{TN}{TN+FP}$ (where TP = true positives, TN = true negatives, FP = false positives and FN = false negatives) are shown at both the nucleotide and exon level. ME is entirely missed exons, WE is wrong exons, and the approximate correlation $AC = \frac{1}{2}(\frac{TP}{TP+FN} + \frac{TP}{TP+FP} + \frac{TN}{TN+FP} + \frac{TN}{TN+FN}) - 1$ summarizes the overall nucleotide sensitivity and specificity by one number. Within each of the three data sets the methods are divided into three classes: those operating on a syntenic DNA pair, those operating on a human sequence using as evidence matches against a database of mouse sequences, and a single-organism gene finder (GENSCAN).

non-coding conserved regions as exons. To test the effectiveness of the CNS state we examined the performance of SLAM on the ROSETTA test set with and without the CNS state. With the CNS state, SLAM predicted 548 CNSs with an average length of 103.2 bp and 78.9% identity. Running SLAM without the CNS state resulted in a drop in nucleotide specificity from 98.1% to 95.4%, and in exon specificity from 75.5% to 69.1%. One would expect the CNS state to increase the rate of false negatives (missed exons) due to the fact that some true exons might be mistaken for CNS, but the exon sensitivity increased, from 76.9% without the CNS state to 78.3% with the state. This can almost certainly be attributed to the protein space exon scoring which effectively distinguishes the type of conservation. Thus, it seems that comparative gene finding requires both an exon boosting component (based on protein alignment) and a conserved non-coding comparison (based on DNA alignment) to be effective.

Another illustrative example demonstrating the importance of the CNS state is the HoxA cluster in human and mouse. The region contains 11 HoxA genes (according to RefSeq annotations), each consisting of two exons. What makes this region particularly difficult for comparative gene finders is the remarkably high level of conservation in both coding and non-coding sequence. The intron and intergenic regions are 69% identical at the DNA level as opposed to about 36% that has been observed on average for human and mouse [22]. This makes the overprediction of exons more likely, particularly for TWINSKAN and SGP-2 which boost exons scores

on the basis of local alignments against a large database. The poor performance is due to the number of false positive exons: 29 for GENSCAN, 53 for TWINSKAN and 31 for SGP-2, as opposed to 10 for SLAM and 10 for TWINSKAN.p.

Figure 1 shows the region of the HoxA2 and HoxA3 genes in human (accession numbers NM_006735 and NM_030661 respectively), where there is a high level of conservation. The TBLASTX hits represent matches between this region and a database of mouse genomic sequence. SLAM and TWINSKAN.p do a good job of distinguishing between exons and CNSs, whereas SGP and TWINSKAN are led to some false positives by the high rate of TBLASTX hits against the mouse genome.

In addition to being a necessary component to reduce false positive predictions, we found that the CNS state enabled the identification of biologically important non-coding features. For instance, we have observed many cases where the SLAM CNS predictions agree excellently with UTR regions (we currently do not report quantitative results for UTR predictions due to the lack of reliable UTR annotations in our data sets).

The model used by SLAM is useful for organism pairs other than human and mouse. It has been retrained for use in comparisons between the Malaria parasites *Plasmodium falciparum* and *Plasmodium vivax*. There are very few sequenced orthologous pairs currently available, but we were interested in these organisms because of the importance of the Malaria genome and the major sequencing efforts underway. Because

of the lack of data we were not able to undertake a performance analysis such as in Table 1 but as an example, we tested on the chlorquin resistance transporter syntenic gene pair (accessions AF030694 and AF314649). **SLAM** correctly found 9 of the exon pairs, there being 13 exons in *P. falciparum* and 14 in *P. vivax*. The performance is good when one considers that the third exon in *P. falciparum* has an intron inserted in *P. vivax*, leading to a differing number of exons and violating the model assumptions. Moreover, the first two exon pairs were not included in the approximate alignment determined by **AVID** because of weak sequence homology, and so were not even considered. Also the smaller size of this example allowed us to test the program with larger approximate alignments, thus moving us closer to simultaneous alignment and gene prediction, and these approximate alignments did not result in any extra false positives.

3. DISCUSSION

SLAM is the first implementation of a GPHMM which simultaneously aligns and predicts genes in two orthologous sequences. Moreover, the requirement of valid gene structures in both sequences improves the accuracy of the program, most notably reducing the false positive rate. The novel components of the program, such as a CNS state and paired exon scoring in protein space to distinguish coding from non-coding conservation, make **SLAM** a powerful tool that can be used for gene prediction as well as for alignment.

SLAM compares favorably to other gene finders, particularly with regards to the false positive rate which has been the Achilles' heel of many gene finding programs. It should be noted that the numbers quoted in our comparisons should be examined qualitatively to determine the relative strengths and weaknesses of the programs, rather than to obtain quantitative measures of their (expected) performance. In the tests performed it was impossible to ensure that the programs were trained and tested on the same sequences, this partly due to the fact that there is not a lot of publicly available, well annotated, orthologous sequence. Furthermore, different programs were optimized for different inputs. For instance, most of the gene finders (including **SLAM**) were optimized for larger genomic regions (or even for draft sequence) rather than single gene sequences such as in the **ROSETTA** set. To account for this we also tested on two long regions, the HoxA cluster and the Elastin region. An extensive and quantitative comparison similar to the single organism gene finding comparison in [20] is a worthwhile endeavor to pursue in the future as more data becomes available.

Nevertheless, we believe that the results obtained shed light on some of the relative strengths and weaknesses of the programs tested, and are valuable in that regard. For example, it is a well known fact that single organism GHMM based gene-finders such as **GENSCAN** and **GENIE** have high false positive rates [20], and it has been universally accepted that "adding" homology information can reduce this problem. However, merely adding alignment information by boosting the scores of highly conserved potential exons is not enough. In shorter, single-gene regions, such as in the **ROSETTA** set, the difference between **GENSCAN** and **TWINSKAN** is negligible. On the other hand, longer, highly conserved regions, such as the HoxA cluster demonstrate the difficulty faced by an

approach that does not explicitly address the problem of distinguishing the type of conservation (the **TWINSKAN** program does leverage the third base pair wobble in determining whether to boost an exon score, but it does not distinguish coding from non-coding conservation). The introduction of a CNS state turned out to be crucial in order to bring down the false positive rate, and we included it in the model only after we discovered that it was impossible to remove false positive predictions of UTRs that were highly conserved, and happened to contain open reading frames.

Examining CNS predictions by eye we have already noticed that they should be valuable for the detection of non-coding features, such as regulatory regions and UTRs. For instance, there were several examples in the test sets where **SLAM** CNS predictions overlapped with annotated UTRs. The CNS lengths and boundaries depend on the parameters, and despite the intuition that conserved regions should be easy to identify, we observed that it was easy to select parameters that either captured, or omitted, various non-coding regions that could be considered to be conserved. Indeed, it appears that predictions of CNSs will be an important application of human mouse comparative studies, and it remains an open problem to establish precise criteria for what a CNS is. The CNS model described in this paper can be extended and enhanced to take advantage of more complex conservation patterns when these are revealed through biological studies.

On the **ROSETTA** set, the **SLAM** performance is somewhat lower at the exon level. However, the high nucleotide sensitivity and specificity in conjunction with a low rate of missed exons indicates that most exons not predicted correctly have a significant overlap with true exons, the exon boundaries being slightly off. Future developments to the **SLAM** program will include the introduction of a paired splice site model. The development of a good theoretical model for scoring splice sites in pairs remains an interesting, unsolved problem. The structure of the underlying Markovian state space in **SLAM** models genes in both organisms, assuming the same number of exons in each, and in the same order. As mentioned above, this is a strength of the model allowing for an increase in specificity by imposing additional constraints which are almost always valid. However, these assumptions are violated (albeit less than 1 percent of the time in human/mouse [23]) and some orthologous genes have different numbers of exons, and/or frameshifts (related exons which have lengths that do not differ by a multiple of 3). These difficulties can be addressed by suitably modifying the GPHMM used. While such modifications will come at a computational cost, by appropriately selecting parameters they should not compromise the accuracy of the program. The generalization of the model to allow for such cases will be particularly valuable for alignment and gene finding between more distant organisms.

A serious issue that arises with the **SLAM** GPHMM is that in a naive implementation the memory usage and the computational complexity scale as the order of the product of the lengths of the input sequences. One way of mitigating this problem is to pre-process the data, producing an approximate alignment as described in the methods section, such that the computational task grows linearly in the length of one of the input sequences. This already helps immensely

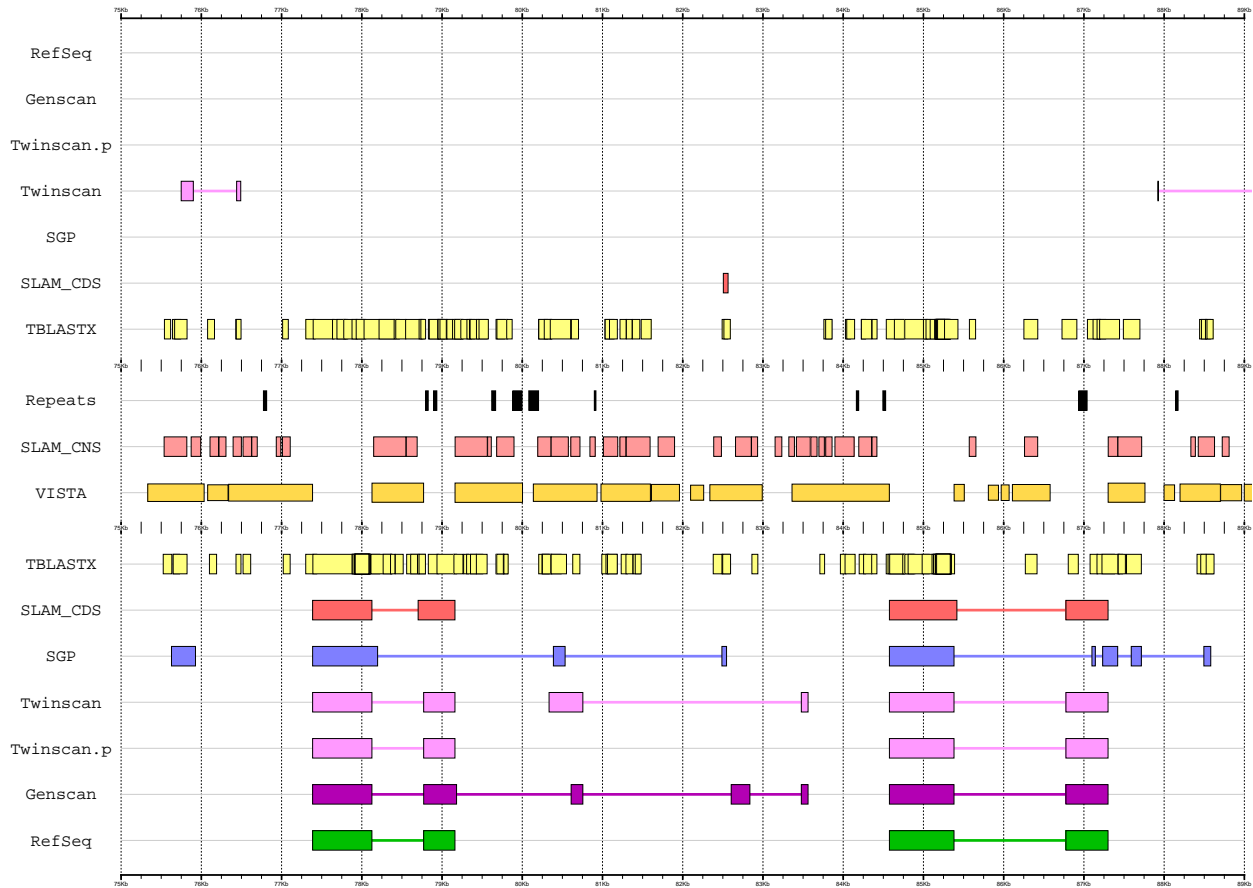


Figure 1: 14,000bp from the HoxA cluster showing the HoxA2 and HoxA3 genes. The top half of the figure consists of predictions and annotations for the 5' → 3' strand and the bottom half for the 3' → 5' strand. The tracks shown are: RefSeq annotations, GENSCAN, TWINSKAN, SGP-2 and SLAM predictions, Repeats masked by RepeatMasker (A. F. A. Smit, and P. Green, unpublished results), TBLASTX alignments, and SLAM and VISTA CNS annotations. The figure was created using gff2ps by J. F. Abril and R. Guigó, available at <http://www1.imim.es/software/gfftools/GFF2PS.html>.

but the pressing need for high-throughput algorithms requires even more sophisticated methods to reduce the memory and computational demands. Producing lean approximate alignments is an interesting problem in its own right, and we have been exploring different strategies, one of which is in development.

The SLAM implementation we have described in this paper is designed for comparing finished sequences. Nevertheless, it is possible to modify the SLAM algorithm for comparison of finished with draft sequence. This functionality is particularly suitable to the current comparison of the human genome with the draft mouse genome. A thorough discussion and analysis of gene finding using finished/draft comparison will appear in another paper; we do however, comment that our results comparing SGP-2 and TWINSKAN (which use mouse whole genome shotgun reads) with TWINSKAN.p

and SLAM (which use syntenic pairs) shows a significantly lower level of specificity for the former due to numerous false alignments leading to false positive predictions. These results confirm the intuition that finished orthologous sequence is imperative for high quality gene predictions, and in fact this issue should motivate sequencing efforts aimed at finishing draft genomes.

A fundamental limitation of the SLAM approach that we have not addressed is the difficulty of rearrangements. The SLAM method assumes that the sequences being compared contain genes and exons that are preserved in order. Unfortunately, certain genome comparisons will show many such evolutionary shuffles. One way to use SLAM in such situations is to pre-process the sequences, for example by chopping them up into segments small enough that rearrangements are no longer an issue. Other evolutionary occurrences, such as

genes within introns, cannot be handled with simple modifications of the GPHMM. Nevertheless, in its current form, **SLAM** is already a useful tool for gene prediction and alignment. It has also been shown to work effectively in other organism pairs — we have retrained **SLAM** on *Plasmodium falciparum* and *Plasmodium vivax* to good effect (though it is interesting to note that processing *Plasmodium* sequences using human/mouse parameters also appears to work reasonably well, indicating that gene conservation and not organism specific parameters, is the most important factor in the **SLAM** gene prediction algorithm).

The prospect of investigating the utility of **SLAM** CNS predictions, as well as the application of **SLAM** to finding alternatively spliced transcripts (by looking, for example, for suboptimal parses) is particularly exciting in light of the many successes that have been obtained by application of the comparative method.

A **SLAM** server, including datasets and additional information, is available at <http://bio.math.berkeley.edu/slam/>.

4. METHODS

Pairs of sequences and their associated gene structures and alignment were modeled using a GPHMM [24]. The input to **SLAM** consists of two sequences and an *approximate alignment* [24]. Approximate alignments are used to reduce the search space for the Viterbi algorithm, and allow for improvements in speed and reductions in memory usage. The main components in the **SLAM** GPHMM are currently an exon boundary detector, an intron/intergene (*I*-state) model, an exon pair scoring model, and a conserved non-coding sequence (CNS) model. The state space and structure of the **SLAM** GPHMM is described below, followed by details of the various new components we have introduced.

4.1 The **SLAM** GPHMM

There are two different kinds of HMMs relevant to our problem: pair HMMs and generalized HMMs. While HMMs generate one single output in each step, a PHMM generates output in pairs, and GHMMs can generate output of different lengths (determined from a distribution) in each hidden state. The **SLAM** GPHMM is a combination of a PHMM and a GHMM. We give a brief overview here of the algorithms needed, more details can be found in [24].

The main difference between the **SLAM** GPHMM model and previous HMM based gene finders is the interpretation of the outputs of the states. The **SLAM** model is a PHMM, and so the outputs in every state are aligned pairs of DNA bases. It is also a GHMM, meaning that there is a duration distribution associated with each of the generalized states (the exon states in this case). The result of combining the two HMMs is that the generalized states now generate *two* sets of durations (or lengths) for the exons, one for each of the sequences.

The state space of the **SLAM** GPHMM is outlined in Figure 2 (the model also contains a mirror-image to the unidirectional model, which allows for finding genes on both the forward and the reverse strands). The generalized states (unshaded) have been distinguished from states which allow self-transitions (shaded) to highlight the resulting partition-

ing of the state space. This partition results in the property that every unshaded state must be followed by a shaded one. This feature allows for a simplification of the HMM algorithms, in particular it is only necessary to compute the various HMM variables for shaded states.

Let $S = \{s_1, \dots, s_N\}$ denote the state space and X_1, \dots, X_L the set of hidden states that the GPHMM follows as it generates the output. With each hidden state we associate a pair of duration times (d_i, e_i) , generated from a joint distribution, and representing the output lengths in that state. We let $p_i = \sum_{k=1}^i d_k$ and $q_i = \sum_{k=1}^i e_k$ denote the partial sums of the durations, and we assume that we have all of the observations in both sequences by the time we reach the final state X_L , or in other words that $p_L = T$ and $q_L = U$, where T, U are the sequence lengths. The variables L, X_1^L, d_1^L and e_1^L are hidden, and we observe only the DNA sequences Y_1^T and Z_1^U (we are using the notation Y_a^b to represent the sub-sequence Y_a, \dots, Y_b).

The two main problems to be solved when applying the HMM theory are: (1) to compute the probability of the observed data, and (2) given the observed sequences, to find the underlying hidden sequence of states that correspond to the optimal alignment and gene structure annotation for the observations. To compute the probability of the observed DNA sequences we calculate the *forward variables*

$$\begin{aligned} \alpha(t, u, i) &\equiv \\ &\equiv \Pr\left(Y_1^t, Z_1^u, \{\text{some hidden state ends in } s_i \text{ at } (t, u)\}\right) \\ &= \Pr\left(Y_1^t, Z_1^u, \cup_{i=1}^{t+u} (X_i = s_i, p_i = t, q_i = u)\right). \end{aligned}$$

where t and u are indices to the first and second sequences respectively. Then, since $\alpha(T, U, i) = \Pr(Y_1^T, Z_1^U, X_L = s_i)$, the probability of the observed data is obtained by

$$\Pr(Y_1^T, Z_1^U) = \sum_{i=1}^N \alpha(T, U, i).$$

The second problem is solved by computing the *Viterbi variables*

$$\delta(t, u, i) = \max_{i, X_1^{i-1}, d_1^{i-1}, e_1^{i-1}} \Pr(Y_1^t, Z_1^u, X_1^{i-1}, X_i = s_i, p_i = t, q_i = u)$$

which are essentially the same as in the forward algorithm with sums replaced by maxima and a little extra book-keeping to keep track of the maximizing terms. The optimal underlying state sequence is then retrieved by backtracking through those maximizing terms. In **SLAM** the forward and Viterbi variables are computed by applying the dynamic programming method.

A key component of the model was the introduction of paired exon states that allow for the computation of exon probabilities based on the alignment in protein space. This is described in more detail below. CNS states were also added, allowing us to model the difference between DNA conservation in introns and intergenes, and protein conservation in coding exons. Splice sites were modeled independently using organism specific, non-stationary *variable length Markov models* (VLMs) as described in [25].

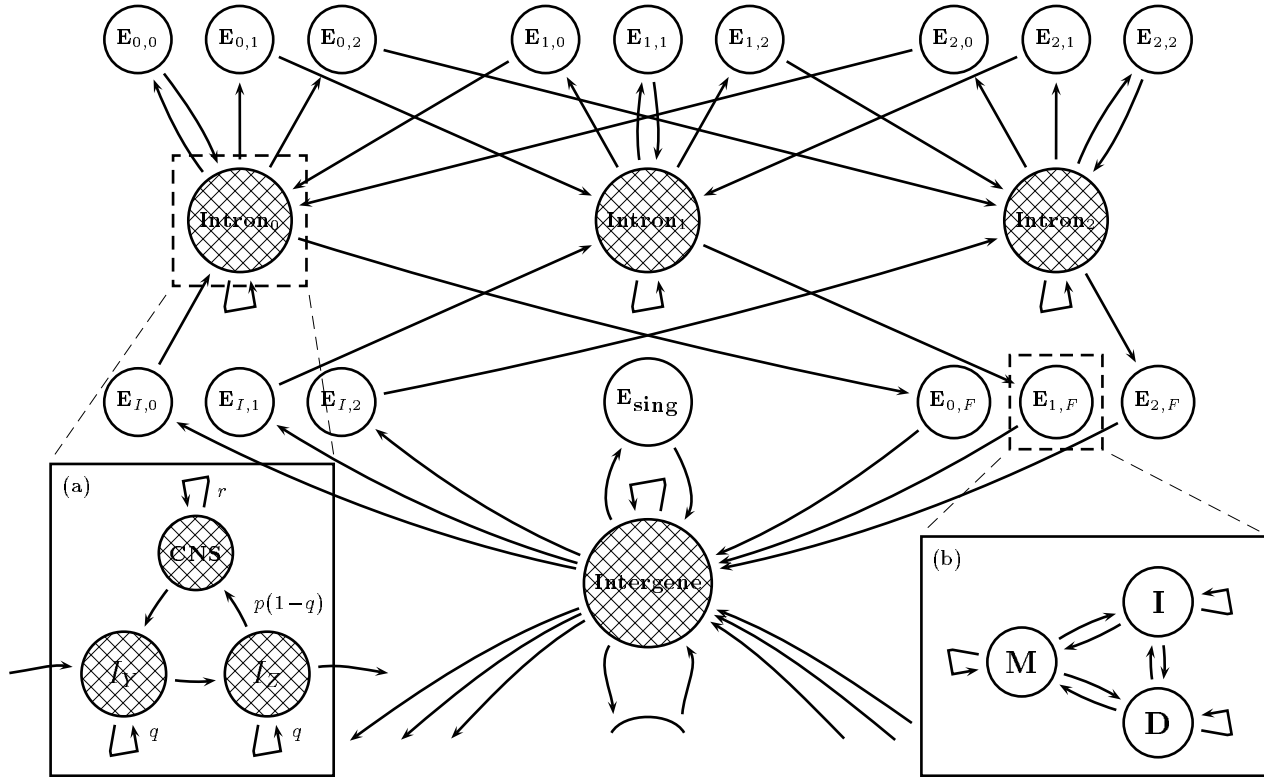


Figure 2: A GPHMM for alignment and prediction of exons using genomic DNA from two different organisms. The shaded states are the typically less-conserved intergene and intron states, each producing either a single base or a gap in each organism. The unshaded states (all of which are exons) will all have duration one as they have no self-transitions, however they are generalized and produce exon-pairs according to some pre-determined joint distribution. (a) In order to avoid the prediction of coding exons in all conserved regions, it was necessary to introduce conserved non-coding states (CNS). Each intron and intergene state consist of two parts: an I -state for modeling long unrelated non-coding regions, and a CNS state for modeling interspersed conserved domains. (b) The modeling of coding exon states in pairs required the construction of a specialized PHMM, consisting of match/mismatch (M), insertion (I) and deletion states (D), which was used to assign probabilities to exon pairs based on alignments in protein space using an appropriate evolutionary model.

4.2 Exon model

A natural probabilistic model for a pair of exons is a PHMM at the amino acid level. However, there are two difficulties with such a model in the context of a gene finding GPHMM: first, the state outputs have to consist of pairs of DNA bases (not amino acids), and secondly, it is necessary to assign probabilities to exon pairs with pre-specified lengths, d and e . From the generative HMM point of view, one cannot assume that the exon pairs are generated by a PHMM which has generated output for some fixed time, since there is no guarantee that the output pairs will have the correct lengths in such a model. Rather, the GPHMM must be thought of as sampling conditionally from the set of all state sequences such that the exons have lengths d and e respectively in the two organisms. In other words, to calculate the probability of an aligned exon pair one has to normalize by the probability that the PHMM produced the given lengths d and e for the respective sequences.

To formalize this, let c_V be the t -th codon coding for an amino acid a_V in an exon sequence V , and let c_W be the u -th

codon coding for amino acid a_W in an exon sequence W . V and W are subsequences of Y and Z , and in what follows it is assumed that they have length divisible by 3; the other cases require simple shifts of the indices using the appropriate mod3 offsets. The forward variables for the exon PHMM are given by

$$\alpha_{\text{exon}}(t, u, i) = \sum_j a_{ji} \Pr(c_V, a_V, c_W, a_W | V_1^{3(t-g_j^V)}, W_1^{3(u-g_j^W)}) \times \alpha_{\text{exon}}(t - g_j^V, u - g_j^W, j)$$

where the summation is over the states *match*, *insertion in V* or *deletion in V*, a_{ji} is the transition probability for $s_j \rightarrow s_i$, and (g_j^V, g_j^W) take on values in $\{(1, 1), (1, 0), (0, 1)\}$ according to whether s_j is the match, insertion or deletion state. The key terms are

$$\Pr(c_V, a_V, c_W, a_W | V_1^{3p}, W_1^{3q}) = \Pr(c_V | V_1^{3p}, W_1^{3q}) \Pr(a_W | a_V, V_1^{3p}, W_1^{3q}) \Pr(c_W | a_W, V_1^{3p}, W_1^{3q})$$

where p, q take on the values of the indices in the recur-

sion for α_{exon} . The probabilities $\Pr(c_V)$ and $\Pr(c_W|a_W)$ were obtained from codon usage tables, and $\Pr(a_W|a_V)$ was obtained from a PAM matrix [26] for an appropriate evolutionary distance (we used a PAM20 for *H. sapiens* vs *M. musculus*). Note that the term $\Pr(a_V|c_V) = 1$. The formula above results in a codon-based PAM matrix (61x61, since stop codons are excluded), determined by using a regular amino acid PAM matrix and marginal distributions of codon usage for the organisms considered. The expression is symmetric in V and W . The dependency on previous sequence in the codon usage tables was modeled with a 5th order Markov model (corresponding to codon pair correlations). In the case of the PAM matrix this dependency was omitted. Gap probabilities for the sequences were obtained from the PAM matrix by summing $\Pr(\text{gap}, aa)$ over all amino acids aa .

The normalization mentioned previously is achieved by excluding the output probability term in the forward variable calculation, *i.e.* by calculating

$$\tilde{\alpha}_{\text{exon}}(t, u, i) = \sum_j a_{ji} \tilde{\alpha}_{\text{exon}}(t - g_j^V, u - g_j^W, j),$$

and dividing α_{exon} by $\tilde{\alpha}_{\text{exon}}$.

4.3 Intron and intergenic models

Simple PHMMs, such as the one shown in Figure 2(b), have the inherent property that any combination of parameters results in a correlation between the lengths of the output sequences. This restrictive property coupled with the empirical observation that in pairs of orthologous sequences, non-coding regions appear to consist of unrelated, non-conserved regions interspersed by highly conserved regions, led us to develop a more refined PHMM for the intron and intergenic states.

The model, shown in Figure 2(a), is formed of two components: the first component, consisting of states I_Y and I_Z , generates a pair of independent intron or intergenic (I -state) sequences, the second component consists of a CNS state for generating related, conserved, non-coding sequence. The I_Y and I_Z states were each modeled as a single state 2nd order Markov model, leading to the generation of independent I -state sequences with geometrically distributed lengths. In addition, the self-transition probabilities for I_Y and I_Z were set to be equal; this was found to be reasonable for human/mouse comparisons. A standard PHMM was used for the CNS state, having the advantage of creating Needleman-Wunsch type DNA alignments for the CNS pairs.

In the absence of a set of well-annotated CNS regions, or even a precise definition of what constitutes a CNS, setting the model parameters is not a straightforward issue. A little circular reasoning was used, whereby initial parameter estimates were used to produce annotations, which were in turn used to guide revised parameter estimates. The parameters are:

$$\begin{aligned} r &= \Pr(\text{CNS} \rightarrow \text{CNS}) \\ p &= \Pr(\text{Intron} \rightarrow \text{CNS} \mid \text{leaving intron}) \\ q &= \Pr(\text{Intron} \rightarrow \text{Intron}) \\ P, Q &= \text{same as } p, q \text{ for intergene} \end{aligned}$$

Requiring that the expected lengths of inter-CNS and entire intron/intergene regions match the average values in the training set leads to four constraints in the five unknowns. To provide a fifth constraint we used an invariance assumption (based on [27]) that it is as likely for a CNS to occur at any given spot in an intron as in an intergenic region, or in other words that

$$p(1 - q) = P(1 - Q).$$

4.4 Computational complexity

A naive implementation of the GPHMM described has the drawback that the Viterbi algorithm has a running time on the order of $D^4 N^2 TU$ where D is the maximum allowable length for an exon (on the order of thousands), N is the number of states, and T and U are the two sequence lengths. The memory requirements are on the order of NTU which also scales as the product of the sequence lengths – ideally we would like the problem to grow linearly in the length of the larger of the observation sequences. Since most alignments in the space of all possible alignments are very unlikely to be real, we adopted the approach of pre-processing to restrict the alignment search space to a set of more likely, or reasonable alignments. We call a set of possible alignments an approximate alignment (details in [24]); this is similar to the concept of the *envelope* of an alignment [17].

Our strategy was to first align the two input sequences using the AVID global alignment tool [21]. AVID is a recursive anchor based alignment algorithm, that generalizes and extends GLASS [23, 4] and MUMmer [28]. The AVID global alignment was “relaxed” in two steps. First by extending the base-to-base alignments to an interval or window of bases surrounding each matching base. We used a window size of 3 bases. Larger window sizes slow the program down and require more memory, but increase the chance that the Viterbi algorithm will find the best (in the sense that orthologous exons will be properly aligned) alignment between the sequences. A window size of 1 would be equivalent to separating the alignment and gene finding steps, as is done in the ROSETTA program [4]. Secondly, the potential state boundaries (*e.g.* boundaries separating exons and introns) were localized and the approximate alignment was expanded around them.

4.5 Parameters

The SLAM GPHMM parameters can be divided into two categories: those parameters that are organism specific, and parameters that depend on the evolutionary distance of the two input organisms. It is interesting to note that in the current implementation of SLAM, only the CNS and exon-pair parameters are in the latter category. The exon states require the selection of an appropriate PAM matrix, and the CNS states require a similar paired output distribution on the DNA level. We selected these parameters by using aligned sequences of the organism pairs in which we were interested.

Initial and transition probabilities, splice site VLMMs, state duration distributions, and output probabilities were all obtained from appropriate training sets. Parameters were stratified by GC content as described in [14]. Parameter sets for different pairs of organisms can be obtained easily with

the SLAM parameter toolbox, which parses GENBANK files containing annotated sequences, generating all the required parameters.

The training sets used for obtaining the results in the paper consisted of the GENIE human set [15]. The same parameters were used for both human and mouse sequences. The parameters were stratified according to GC content into four bins: bin1 = [0,43], bin2 = [43,51], bin3 = [51,57], bin4 = [57,100]. As an example, in bin2 $t_1 = 1320$, $t_2 = 838$, $x_1 = 1103$ and $x_2 = 741$ resulting in $P \approx 0.151$ and an average CNS length of 119.2.

Finally, the output distribution in the CNS state was set such that each pair of bases was independently generated from a joint distribution over $\{A,C,G,T\} \times \{A,C,G,T\}$ where the probability of a match was set to 0.5, the distribution being otherwise uniform.

5. ACKNOWLEDGMENTS

We thank Terry Speed and David Kulp for helpful suggestions and support, and James Harly Gorrell for technical computing advice. M. A. was supported by STINT, the Swedish Foundation for International Cooperation in Research and Higher Education.

6. REFERENCES

- [1] Pennacchio, L. A., Olivier, M., Hubacek, J. A., Cohen, J. C., Cox, D. R., Fruchart, J., Krauss, R. M. & Rubin, E. M. (2001) *Science* **294**(5540), 169–173.
- [2] Hardison, R. C., Oeltjen J. & Miller, W. (1997) *Gen. Res.* **7**(10), 959–966.
- [3] Lander, E. S. (1998) *Personal communication*.
- [4] Batzoglou, S., Pachter, L., Mesirov, J., Berger, B. & Lander, E. S. (2000) *Gen. Res.* **10**(7), 950–958.
- [5] Bafna, V. & Huson, D. H. (2000) *ISMB-00: Proceedings of the Eighth International Conference on Intelligent systems for Molecular Biology*.
- [6] Korf, I., Flicek, P., Duan, D. & Brent, M. R. (2001) *Bioinformatics* **1**(1), S1–S9.
- [7] Wiehe, T., Gebauer-Jung, S., Mitchell-Olds, T. & Guigó, R. (2001) *Gen. Res.* **11**(9), 1574–1583.
- [8] Guigó, R. (2001) *Personal communication*.
- [9] Gelfand, M. S., Mironov, A. A. & Pevzner, P. A. (1996) *Proc. Natl. Acad. Sci. USA* **93**, 9061–9066.
- [10] Yeh, R. F., Lim, L. P. & Burge, C. B. (2001) *Gen. Res.* **11**(5), 803–816.
- [11] Birney, E. & Durbin, R. (2000) *Gen. Res.* **10**(4), 547–548.
- [12] Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990) *J. Mol. Bio.* **215**, 403–410.
- [13] Kent, W. & Zahler, A. (2000) *Gen. Res.* **10**(8), 1115–1125.
- [14] Burge, C. & Karlin, S. (1997) *J. Mol. Bio.* **268**, 78–94.
- [15] Reese, M. G., Kulp, D., Tammana, H. & Haussler, D. (2000) *Gen. Res.* **10**(4), 529–538.
- [16] Needleman, S. B. & Wunsch, C. D. (1970) *J. Mol. Bio.* **48**, 443–453.
- [17] Holmes, I. (1998) *Ph.D. Thesis*, University of Cambridge and Sanger Centre, UK.
- [18] Durbin, R., Eddy, S., Krogh, A. & Mitchison, G. (1998) *Biological sequence analysis*. Cambridge University Press.
- [19] Elnitski, L., Miller, W. & Dewar, K. (2001) *Personal communication*.
- [20] Bures, M. & Guigó, R. (1996) *Genomics*, **34**, 353–357.
- [21] Bray, N., Schwartz, J., Lord, J., Dubchak, I. & Pachter, L. (2001) <http://bio.math.berkeley.edu/avid/>.
- [22] Makalowski, W., Zhang, J. & Boguski, M. S. (1996) *Gen. Res.* **6**(9), 846–857.
- [23] Pachter, L. (1999) *Ph.D. thesis*, Department of Mathematics, Massachusetts Institute of Technology.
- [24] Pachter, L., Alexandersson, M. & Cawley, S. (2001) *RECOMB 2001: Proceedings of the Fifth International Conference on Computational Molecular Biology*.
- [25] Cawley, S. (2000) *Ph.D. Thesis*, Department of Statistics, U.C. Berkeley.
- [26] Dayhoff, M. O., Schwartz, R., Orcutt & B. C. (1978) in *Atlas of Protein Sequence and Structure*, ed. Dayhoff, M. O. (Natl. Biomed. Res. Found., Washington, DC), Vol. 5, Suppl. 3, 345–352.
- [27] Bergman, C. M. & Kreitman, M. (2001) *Gen. Res.* **11**(8), 1335–1345.
- [28] Delcher, A. L., Kasif, S., Fleischmann, R. D., Peterson, J., White, O. & Salzberg, S. L. (1999) *Nucl. Ac. Res.*, **27**(11), 2369–2376.