BIOINFORMATICS APPLICATIONS NOTE
Vol. 20 no. 16 2004, pages 2878–2879
doi:10.1093/bioinformatics/bth315

TigrScan and GlimmerHMM: two open source ab initio *eukaryotic gene-finders*

W. H. Majoros, M. Pertea and S. L. Salzberg*

Bioinformatics Department, The Institute for Genomic Research, Rockville, MD 20850, USA

Received on April 1, 2004; revised on April 28, 2004; accepted on May 3, 2004
Advance Access publication May 14, 2004

ABSTRACT

Summary: We describe two new Generalized Hidden Markov Model implementations for *ab initio* eukaryotic gene prediction. The C/C++ source code for both is available as open source and is highly reusable due to their modular and extensible architectures. Unlike most of the currently available gene-finders, the programs are re-trainable by the end user. They are also re-configurable and include several types of probabilistic submodels which can be independently combined, such as Maximal Dependence Decomposition trees and interpolated Markov models. Both programs have been used at TIGR for the annotation of the *Aspergillus fumigatus* and *Toxoplasma gondii* genomes.

Availability: Source code and documentation are available under the open source Artistic License from http://www.tigr.org/software/pirate.

Contact: bmajoros@tigr.org

INTRODUCTION

With the increased availability of raw genomic sequence data has come an increase in the number of gene-finder programs available for predicting the protein-coding genes in these data. Unfortunately, the vast majority of these programs cannot easily be retrained by end users, because these packages rarely include retraining software, and in most cases the source code is not available, which also limits modification and reuse of these programs for functionally different annotation tasks.

We describe two new gene finders, GlimmerHMM and TigrScan, which are based on the same class of models as Genscan (Burge, 1997) and Genie (Kulp *et al*., 1996), namely, a Generalized Hidden Markov Model (GHMM). GHMMs offer the advantage of providing a probabilistically rigorous framework in which alternative gene-finding strategies can be readily explored. Furthermore, since our source code is available as open source, and because the programs are written in a highly modular C/C++ style, reusing portions of the programs for novel annotation tasks is made quite feasible.

METHODS

A Hidden Markov Model (HMM) is a state-based generative model which transitions stochastically from state to state, emitting a single symbol from each state according to that state's emission probabilities. A GHMM generalizes this process by emitting complete gene features, or subsequences, in each state. Because each state can be associated with a different gene feature type (e.g. donor, exon, etc.), a GHMM provides an intuitive and flexible framework for exploring alternative gene-finding approaches. For example, feature states can be independently retrained, and different types of submodels (e.g. Markov models, weight matrices, etc.) can be used at each state. Predicting gene models with a GHMM involves finding the most probable path, ϕ, through the GHMM topology given the sequence, S; i.e. maximizing $P(\phi|S)$. Bayes' theorem and the invariance of the marginal probability $P(S)$ with respect to individual paths ϕ gives:

$$\begin{aligned}\operatorname*{argmax}_{\phi} P(\phi|S) &= \operatorname*{argmax}_{\phi} \frac{P(\phi \wedge S)}{P(S)} = \operatorname*{argmax}_{\phi} P(\phi \wedge S)\\ &= \operatorname*{argmax}_{\phi} P(S|\phi)P(\phi).\end{aligned}$$

Because the GHMM allows explicit modeling of state duration (feature length) d_i for each state q_i in parse ϕ, this can be factored into

$$\operatorname*{argmax}_{\phi} \prod_{q_i \in \phi} P(S_i|q_i \wedge d_i)P(q_i|q_{i-1})P(d_i),$$

where $P(q_i|q_{i-1})$ is the probability of transitioning from state q_{i-1} to q_i, S_i is the subsequence emitted by state q_i, and $P(d_i)$ is the probability of state q_i emitting a feature of length d_i. These can all be estimated from training data through various well-known means (e.g. Salzberg *et al*., 1998). This optimization step can be efficiently evaluated using a dynamic programming approach.

Though both TigrScan and GlimmerHMM conform to the overall mathematical framework of a GHMM, they differ significantly from each other and from our previous gene finders in the details of their implementation—specifically, in the

*To whom correspondence should be addressed.

Table 1. Results on a set of 800 full-length *Arabidopsis thaliana* cDNAs (*a.t.*) and on 360 curated *Aspergillus fumigatus* CDSs (*a.f.*)

	% Nucl. accuracy		% Exon sensitivity		% Exon specificity		% Exact genes	
	a.t.	*a.f.*	*a.t.*	*a.f.*	*a.t.*	*a.f.*	*a.t.*	*a.f.*
TigrScan	96	90	77	37	81	47	43	19
GlimmerHMM	96	91	71	36	79	49	33	21
Genscan+	95	87	75	23	82	4	35	11

Exon sensitivity = TP/(TP+FN), where TP stands for true positives and FN for false negatives, with a TP indicating that both exon coordinates were correct. Exon specificity = TP/(TP+FP), where FP stands for false positives. Exact genes is the percentage of the test CDSs for which the predictions were entirely correct. Genscan+ is an *A.thaliana* specific version of Genscan provided to us by C. Burge.

statistical methods employed at the submodel level and in their overall software architecture. Whereas TigrScan utilizes several types of weight matrices and Markov chains, GlimmerHMM additionally incorporates splice site models adapted from the GeneSplicer program (Pertea *et al.*, 2001) and a decision tree adapted from GlimmerM (Salzberg *et al.*, 1999). Both programs utilize interpolated Markov models (Salzberg *et al.*, 1999) as well as the Maximal Dependence Decomposition technique for improving specificity in splice site identification (Burge, 1997). Currently, TigrScan's GHMM structure includes introns of each phase, intergenic regions, 5′- and 3′-untranslated regions (5′- and 3′-UTRs), and four types of exons (initial, internal, final, and single). GlimmerHMM includes states for exons, introns and intergenic regions.

TigrScan also provides as an optional feature the construction of a graph-theoretic representation of all high-scoring open reading frames. Such graphs have been found to be useful in several ongoing research projects, including a homology-based gene finder as well as two other projects which explore unconventional approaches to genomic annotation. TigrScan can also read and score an arbitrary gene model provided in GFF format. These features allow us to dynamically explore the immense space of suboptimal gene models in ways that are simply not possible with most other gene finders.

RESULTS

Both programs performed well in tests when compared with Genscan+ (Table 1). Of the three gene finders, TigrScan was found to perform most competitively on the *A.thaliana* test set for three of the four reported measures, whereas GlimmerHMM was found to perform best on the *A.fumigatus* test set for three of the measures. The greater difference in accuracy between our gene finders and Genscan+ on the *A.fumigatus* set demonstrates the value of being able to retrain the gene finders for specific organisms.

Table 2. Memory and time requirements on a 922 kb *A.fumigatus* contig

	Memory (Mb)	Time (min)
GlimmerHMM	84	0:17
TigrScan	29	1:28
Genscan+	445	2:57

Time and memory requirements of both programs increase linearly with the length of the input sequence, though the two programs make different trade-offs between speed and space, as can be seen from Table 2. TigrScan successfully processed a 5.6 Mb contig in 5 min 32 s using 105 Mb of RAM on a 1.6 GHz Pentium IV, illustrating that long sequences can be processed even on machines with relatively limited memory.

By offering both these programs to the community as open source, we hope to facilitate more studies comparing the suitability of alternate gene-finding strategies.

ACKNOWLEDGEMENTS

We would like to thank C. Burge for kindly providing a version of Genscan+ trained for *A.thaliana*, and the anonymous reviewers for their constructive comments which improved this paper. This work was supported in part by NIH grant R01 LM06845 and NSF grant MCB-0114792.

REFERENCES

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

Kulp,D., Haussler,D., Reese,M. and Eeckman,F. (1996) A generalized hidden Markov model for the recognition of human genes in DNA. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **4**, 134–142.

Pertea,M., Lin,X. and Salzberg,S.L. (2001) GeneSplicer: a new computational method for splice site prediction. *Nucleic Acids Res.*, **29**, 1185–1190.

Salzberg,S.L., Searls,D.B. and Kasif,S. (eds) (1998) *Computational Methods in Molecular Biology*. Elsevier, Amsterdam, The Netherlands.

Salzberg,S.L., Pertea,M., Delcher,A.L., Gardner,M.J. and Tettelin,H. (1999) Interpolated Markov models for eukaryotic gene finding. *Genomics*, **59**, 24–31.