Published online 14 December 2006 *Nucleic Acids Research, 2007, Vol. 35, No. 1* ***317–324***
doi:10.1093/nar/gkl1026

SURVEY AND SUMMARY

An evaluation of contemporary hidden Markov model genefinders with a predicted exon taxonomy

Keith Knapp[1,*] and Yi-Ping Phoebe Chen[1,2,*]

[1]Faculty of Science and Technology, Deakin University, Australia and [2]Australia Research Council Centre in Bioinformatics, Australia

Received September 21, 2006; Revised and Accepted November 13, 2006

ABSTRACT

We present an independent evaluation of six recent hidden Markov model (HMM) genefinders. Each was tested on the new dataset (FSH298), the results of which showed no dramatic improvement over the genefinders tested five years ago. In addition, we introduce a comprehensive taxonomy of predicted exons and classify each resulting exon accordingly. These results are useful in measuring (with finer granularity) the effects of changes in a genefinder. We present an analysis of these results and identify four patterns of inaccuracy common in all HMM-based results.

INTRODUCTION

Since the first genefinding algorithms such as TESTCODE (1) came onto the scene, their effectiveness have grown with nucleotide sensitivity and specificity now reported in the high 90% range (2,3). Despite such nucleotide level results, exon, gene and whole-genome level results still need improvement (4,5). Research presses on towards improving the capabilities of automated gene annotation on the exon and whole-gene levels. Common among putatively 'high' performance genefinders is the implementation of hidden Markov model (HMM) variants. We present an analysis and review of how the contemporary HMM genefinders: Augustus, Genezilla, GenomeScan, GlimmerHMM, SNAP and Twinscan fared on a new dataset (FSH298). Building on this, we apply our novel and comprehensive taxonomy of predicted exons to the output of each program tested. The purpose of this is to identify patterns of inaccuracy common to all HMM genefinders. Subsequently each pattern of inaccuracy can then be addressed hopefully resulting in more accurate genefinders. As this paper specifically evaluates HMM genefinders, a brief review is first provided.

HIDDEN MARKOV MODELS

Genefinders are commonly divided into two categories either *ab initio* or homology based (6,7). As will be discussed, many genefinders are hybrids. Given a sequence of inputs and a set of classes, a HMM assigns a class to each individual input. In the case of genefinders, the inputs are DNA nucleotides and the classes assigned are content signals or other regions, such as exons, introns, Poly(A) tails and TATA boxes. HMMs can be quite effective and have been implemented in other areas such as speech and gesture recognition, DNA and protein homology searches, and genefinding systems (8–10).

As a sequence passes through the HMM genefinder, a class is assigned to each input based on a particular probability associated with the current state. The model itself is a collection of states (each containing an output probability for each class) and the transitions between states, where each transition has a probability of happening. As each input progresses into the model, the focus of the model transitions from its current state to another state (each having a different set of output probabilities).

Multiple extensions of HMMs exist; two common ones are the Generalized HMM and the Pair HMM. In a standard HMM only one input is classified in a given state, afterwards the model must perform a state transition, and the next input is then classified. Generalized HMMs remove the constraint of only one classification per state, and allow multiple classifications (also known as emissions or observations) to occur. As its name suggests pair HMMs compare two nucleotides from separate sequences concurrently and at each state a pair of nucleotides is used in determining which probability is used when emitting a class.

An HMM is composed of five items, some already mentioned. (i) A set of N states, i.e. TATA box, exon, intron, etc. (ii) A set of M observations/classes. (iii) A state transition probability distribution $A = \{a_{ij}\}$. (iv) An observation symbol probability distribution in state j. (v) An initial state distribution π (11).

Associated with HMMs are three problems that must be solved. (i) Evaluation, the probability of a set of observations occurring given a particular HMM. Probabilities like this provide a score on how applicable the given model is to the sequence. The forward–backward algorithm calculates this score (8). (ii) Decoding, determining the optimal order of hidden states to generate the observed sequence.

*To whom correspondence should be addressed. Tel: +61 3 92517684; Fax: +61 3 92517604; Email: phoebe.chen@deakin.edu.au
*Correspondence may also be addressed to Keith Knapp. Tel: +61 3 52272606; Fax: +61 3 52272167; Email: kdk@deakin.edu.au

Commonly employed to solve this is the Viterbi algorithm (12). (iii) Learning, estimating the probability of starting in a given state. No one particular algorithm solves this optimally, but multiple algorithms are used. For example Baum–Welch and Expectation–Maximization algorithms both have been employed to solve this problem (8).

Note that in the genefinding realm it is common practice to identify only four 'exons' (single, initial, internal and terminal); all of which must exist between the start and stop codons. Classifications such as this are simplistic and inexact. It is simplistic in that only four out of twelve (5) possible exon types have ever been considered. It is inexact to define an exon simply as a DNA-coding sequence between two introns because exons also exist in untranslated regions (UTRs).

For genefinders to attain consistently high output this practice must end. Unfortunately bias inherent in the gene research focuses mainly on protein coding exons, and insufficient data exist for training genefinders to annotate non-coding exons (M. Zhang, unpublished data).

METHODS

Due to their pervasiveness in the genefinder market, HMM-based genefinders are the focus of this research. We test six genefinders: Augustus, Genezilla, GenomeScan, GlimmerHMM, SNAP and Twinscan. Two of these (Twinscan and GenomeScan) also employ homology by default for prediction.

Three criteria were used for considering a genefinder for this evaluation. The first and most obvious criteria is that the genefinder must be HMM based on or else an extension to the HMM concept. Additional selection factors were age and testability. We looked for genefinders made available since 2001 which had not been involved in a comparison project similar in nature.

Attempts were made to include Doublescan either version 1 (2) or version 2 beta (I. M. Meyer, unpublished data), but testing could not be completed. Version 1.0 (hosted, but no longer supported at http://www.sanger.ac.uk/Software/analysis/doublescan/) never returned any results. Beta version 2.0 (unpublished data, correspondence with Meyer) suffers from modular dependency issues and only functioned sporadically. TWAIN (13) was considered and then removed, as it already implements a version of GeneZilla and implements homology.

FSH298 DATASET

For the purposes of this test a new dataset, FSH298, was built (available as Supplementary Data at NAR online). The dataset was extracted based on three search criteria:

- The sequence contained a complete CDS.
- The sequence was from human DNA.
- The sequence was published after July 2005.

The publishing date was ensured to be accurate by using the 'limit' feature of Entrez Nucleotide at NCBI and selecting 'Publication Date' from the appropriate drop-down menu.

This ensures that the programs were not trained on sequences in the FSH298 dataset. In addition to ensuring non-overlap of training and test data, we created this dataset to be both testable and heterogeneous than previous tests. It is testable in the sense that we have known CDS annotations, yet is wild in so much as extreme filtering methods were not applied. Unlike previous test sets (18 and 20) we purposefully did not search for or remove sequences with the following characteristics:

- Non-canonical translation start and stop codons (ATG–TAA, TAG, TGA).
- Non-canonical intron boundaries (GT–AG).
- Protein coding frames not evenly divisible by three.

FSH298 has the following properties:

- It consists of 37 genes with no introns in the open reading frame (commonly referred to as a 'single exon gene') and 261 multi-exon genes. The mean number of coding exons per gene is 8.57.
- There are 2555 coding exons with a mean length of 171 bases. There are 2257 introns with a mean length of 3534 bases.
- It consists of 10 793 400 nt over 298 sequences with a mean sequence length of 36 219 bases.
- Four percent of the dataset are CDS, 74% intronic (between coding exons only, not UTR introns), while 22% is neither protein coding nor intronic (thus intergenic, promoter, UTR, Poly(A), etc.).

Regarding alternative splicing only two sequences (DQ070893 and AF479645) in FSH298 had an alternative coding sequence. These alternatives were identified by the GenBank feature tag 'CDS'. The statistics for these two sequences were calculated manually, selecting the alternative with the best match, and integrated with the non-alternative spliced statistics. It is possible that in the future additional CDS features may be annotated which are currently unknown.

TEST PROGRAMS

The following section provides a brief introduction to each program, its training set, output format and other relevant characteristics.

Augustus

This program (14) is originally a generalized HMM for eukaryotes, and has since been expanded to model introns more accurately and to incorporate user-defined heuristics. The original training set used on Augustus was from GenBank of October 2002. It consisted of 1284 single gene sequences. Augustus output is in the GFF format scoring both strands of DNA and assigning a score to each predicted coding sequence. We tested all the sequences of the FSH298 dataset locally on Augustus v.1.5.

GeneZilla

Formerly known as Tigrscan, GeneZilla (15) implements a GHMM. GeneZilla is a mammoth system which capitalizes on the modularity of HMMs. As with all genefinders the author attempted to use training data provided by the developer. With the exception of four submodels: initial-exon, internal-exon, terminal-exon and single-exon, Genezilla was trained on human-models-refseq8000.tar.gz (available from

http://ftp.bioinformatics.org/pub/genezilla). The four submodels were not included with the distribution and were derived from the developer recommended (W. H. Majoros, personal communication) Homo.sapiens.tar.gz dataset (available from http://www.genefinding.org/datasets.html). GeneZilla correctly ran on all 298 sequences.

GenomeScan

Building on the strengths of Genscan, we ran the GenomeScan (3) web server on all FSH298 sequences, and it completed successfully on 294. We ran a BLASTX query on the entire FSH298 dataset in the organism domain Rodentia to obtain homologous data to run in GenomeScan along with FSH298. We selected the highest scoring BLASTX result that was not classified as experimental. GenomeScan output was return by email, the contents of which were copied into text files for processing.

GlimmerHMM

GlimmerHMM v.2.1 (15) is a GHMM for identifying genes on eukaryotes. The dataset used for training GlimmerHMM was assembled in 2004 (M. Pertea, personal communication). Given the full test set GlimmerHMM predicted genes on all sequences. GlimmerHMM output is a proprietary format similar to GFF and was placed into a single text file for parsing and calculating statistics.

SNAP

SNAP's (version 2004-03-02) (16) original focus was to annotate genomes for which gene finder has not yet been fine-tuned. SNAP was first trained on *Arabidopsis thaliana* of the four datasets available from the developer's website (http://homepage.mac.com/iankorf/). SNAP was then retrained on human DNA. The dataset used was the 804 plus strand sequences from the Homo.sapiens.tar.gz dataset (available from http://www.genefinding.org/datasets.html); a subset of those used in partial retraining of GeneZilla. All sequences were tested locally, with a result returned for each sequence.

Twinscan

A pair of HMM, Twinscan (17) implements both second- and fifth-order homogenous Markov chains in gene finding along with mouse homology information. We successfully ran 295 sequences from FSH298 on the Twinscan webserver (available at http://genes.cs.wustl.edu/twinscan). Output of Twinscan was received by email in gtf formatted files.

Statistics

In order to calculate metrics useful for comparing each genefinder we first calculated the following four metrics:

- True positive (TP), a nucleotide that is correctly annotated as coding.
- True negative (TN), a nucleotide that is correctly annotated as a non-coding.
- False positive (FP), a nucleotide incorrectly annotated as coding.
- False negative (FN), a nucleotide incorrectly annotated as non-coding.

Once completed these serve as the basis for the next step, calculation of the following comparison measures.

Nucleotide specificity (NSp) is defined as the proportion of nucleotides that are truly coding:

$$\mathrm{SP} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Nucleotide sensitivity (NSn) is defined as the proportion of annotated nucleotides that are correctly predicted as coding (2).

$$\mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Sensitivity shows a proportion in relation to reality, while specificity shows a proportion in relation to the prediction. Neither Sn nor Sp alone is a good indication of the prediction accuracy because if one has a high value the other may not. A good discussion of this is available in (18). To overcome this issue the following nucleotide measures calculate a value useful for comparisons:

(i) Correlation coefficient (CC) displays a relationship between sensitivity and specificity, when both coding and non-coding regions exist in the training and test datasets.

$$\mathrm{CC} = \frac{\mathrm{TP} * \mathrm{TN} - \mathrm{FN} * \mathrm{FP}}{\sqrt{(\mathrm{TP} + \mathrm{FN}) * (\mathrm{TN} + \mathrm{FP}) * (\mathrm{TP} + \mathrm{FP}) * (\mathrm{TN} + \mathrm{FN})}}$$

(ii) Burset and Guigó (18) introduced the average correlation (AC), derived in part from the average conditional probability (ACP). AC partially resolves the CC deficiency of: a zero value occurring as a factor in the denominator causing a square root of zero calculation error.

$$\mathrm{AC} = (\mathrm{ACP} - 0.05) * 2$$

$$\mathrm{ACP} = \frac{1}{4}\left(\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} + \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} + \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}\right)$$

Unfortunately ACP has a minor, yet previously undocumented drawback. In situations where the test sequence consists of a single exon gene and the genefinder is 100% accurate, AC is unable to calculate and display such precision, as ACP requires the identification of non-coding nucleotides. Lack of non-coding nucleotides causes a division by zero error and the resultant inability to rate a genefinder's accuracy as perfect. Such deficiencies can be overcome by either ensuring that the test data have non-coding nucleotides (e.g. in flanking regions) or programmatically by testing for said conditions and implementing a proper work-around.

Regarding exon level accuracy specificity (ESp) is defined as the proportion of exons that are actually coding. Likewise, exon sensitivity (ESn) is the proportion of exons that are correctly predicted as coding

$$\mathrm{ESp} = \frac{\mathrm{TE}}{\mathrm{PE}} \quad \mathrm{ESn} = \frac{\mathrm{TE}}{\mathrm{AE}}$$

where TE (true exons) is the number of correctly predicted exons, AE (actual exons) is the number of annotated

Table 1. Performance of six Genefinders on the FSH298 dataset

	Nucleotide No genes	SN	SP	AC	CC	Exon CR	PC	OL	ME	WE	SNE	SPE	AVG
Twinscan	7	0.90	0.95	0.89	0.88	0.50	0.34	0.07	0.12	0.07	0.59	0.51	0.55
GenomeScan	43	0.88	0.83	0.72	0.81	0.63	0.07	0.01	0.26	0.14	0.76	0.74	0.75
GlimmerHMM	9	0.89	0.79	0.80	0.80	0.61	0.13	0.03	0.14	0.21	0.69	0.63	0.66
Augustus	0	0.81	0.78	0.78	0.76	0.63	0.12	0.01	0.15	0.17	0.64	0.63	0.64
GeneZilla	0	0.70	0.67	0.67	0.65	0.40	0.16	0.05	0.17	0.31	0.47	0.40	0.44
SNAP (*H.sap*)	9	0.72	0.71	0.69	0.66	0.35	0.20	0.08	0.31	0.34	0.40	0.36	0.38
SNAP (*A.thal*)	7	0.47	0.22	0.22	0.19	0.04	0.10	0.09	0.52	0.76	0.11	0.04	0.08

The metrics provided are for the whole genome, the nucleotide and the exon level. At the whole genome (No genes) is the number of sequences where no gene was predicted. At the nucleotide level sensitivity (SN), specificity (SP), approximate correlation (AC) and the correlation coefficient (CC) are displayed. On the exon level correct exons (CR), partially correct (PC), overlapping exons (OL), missed exons (ME), wrong exons (WE), exon sensitivity (SNE) exon specificity (SPE), and the mean average (AVG) of SNE and SPE. All genefinders successfully completed each of the 298 sequences except Twinscan and GenomeScan which completed 295 and 294, respectively. SNAP was trained on two organisms, *A.thaliana* (*A.thal*) and *H.sapiens* (*H.sap*).

exons and PE (predicted exons) is the number of predicted exons (19).

Finally we calculated the mean average of ESn and ESp. This average places equal weight on both measurements, but consolidates them into a single numerical metric, which has become a *de facto* standard for measuring exon level accuracy.

RESULTS

In obtaining the results of the study we prepared a new dataset FSH298, this consists of genetic sequences added to GenBank after the publication of the training data of the six genefinders in this study. We calculated two sets of results. The first was the traditional measurements for estimating the effectiveness of genefinders (discussed above). These results are in Table 1. The second is the predicted exon taxonomy (PET) described later.

All sequences were attempted on each genefinder; however, in the cases of GenomeScan and Twinscan, four and three sequences failed to complete on each genefinder, respectively. Sequence length caused the failure in Twinscan. GenomeScan however never completed one sequence, despite multiple attempts, and repeatedly returned a stack execution error for three additional sequences. Furthermore, these were the only two which were run via a web interface; all other programs were run locally.

For the purposes of GenomeScan each sequence in FSH298 was BLASTX'ed (20) employing the organism subset Rodentia for comparison. The same organism was used in the BLASTN search to find homologs as input for Twinscan.

In order to confirm correct annotation only results from the positive strands were considered. If multiple genes were predicted on a single sequence, all predicted exons were treated as part of one gene.

Predicted exon taxonomy statistics

The second set of results obtained was the classification of each exon predicted as coding and determining the resulting trends as presented in the following section. No genefinder evaluations has until now presented such a comprehensive taxonomy of all possible exon classifications. Burset and Guigó (18) measured 'Missed Exons' and 'Wrong Exons'. Rogic, Mackworth and Ouellette (19) extended this with 'Partially Correct' and 'Overlapping exons'.

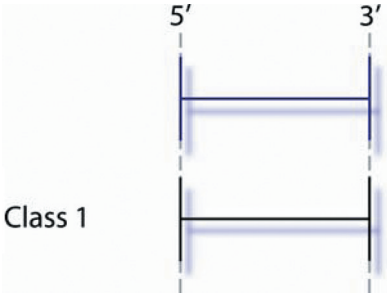

Figure 1. Class 1 exons. Match exactly at both boundaries.

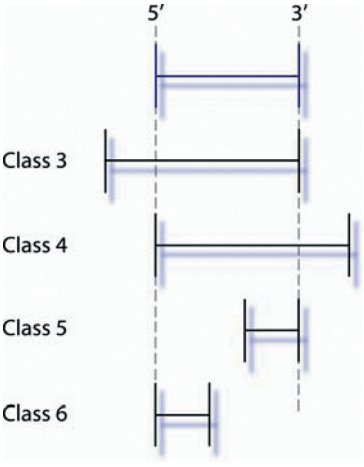

Figure 2. Partially correct. Classes 3–6 match only one boundary.

Each exon predicted was classified into one out of thirteen categories, and further identified by each genefinder as initial, internal, terminal or single. These are illustrated in Figures 1–4. Class 1 exons are correct on both boundaries (Figure 1). Classes 2–9 cover every possible type of overlap. Rogic *et al.* (19) separate these into two categories, partially correct and overlapping. The former have one boundary correct (Figure 2), and include classes 2, 7, 8 and 9.

The latter (Figure 3) match no boundaries, and are composed of classes 3–6. Exons of classes 10 and 11 either end on an annotated 5′ boundary or start on a known 3′ boundary (Figure 4). These should never occur, however if an exon of this type occurs, it may well warrant further investigation.

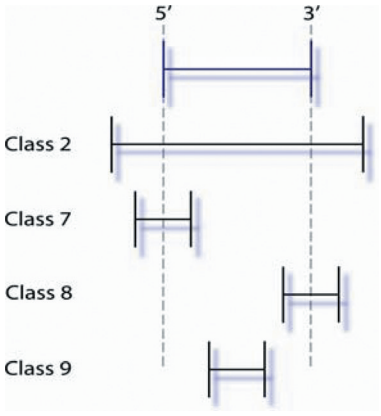

Figure 3. Overlapping exons. No boundaries match but the exons do overlap true annotated exons for classes 2, 7, 8 and 9.

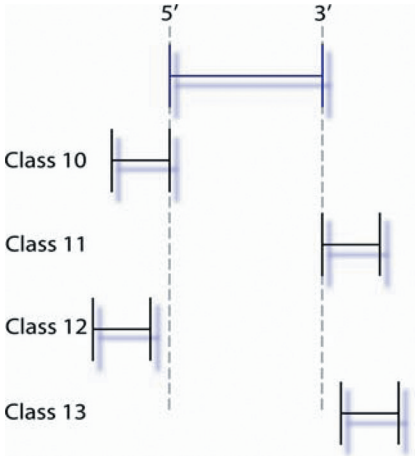

Figure 4. Wrong exons. Classes 10 and 11 reverse a boundary. Classes 12 and 13 neither match a boundary nor overlap an annotated exon.

Classes 12 and 13 are both wrong exons, neither touching nor overlapping any annotated boundary. Their class is solely determined by their distance in bases to the nearest actual exon boundary (upstream or downstream).

The name PET is especially correct as genefinders should identify all exons (whether non-, partially- or fully-coding). The PET is applicable to all predicted exons, not just the coding ones.

Such a formal taxonomy of predicted exons (as the one presented) is necessary for multiple reasons. The first is to identify patterns of incorrect exon identification common to all HMM-based genefinders. By using the class and the feature (initial, internal, terminal or single exons) we can identify patterns at the finest level possible. Thus new genefinders can be engineered to resolve such issues.

The second reason is to see the direct result of changes to a particular genefinder. Generation of PET statistics clearly show which classes a genefinder tends to predict most.

In order to classify a predicted exon, a reference exon must first be found. These are found by comparing the predicted exon to every annotated exon for the sequence in question. An annotated exon is a candidate to be a reference exon if any bases overlap the predicted exon. The annotated exon with the largest number of overlapping bases is then selected as the reference exon. If no annotated exons overlap the predicted exon, then the exon closest physically is the reference exon. The predicted exon is classified based on its 5′ and 3′ boundaries in relation to the selected reference exon.

Twinscan in addition to identifying exons annotates 'start_codons' and 'stop_codons' as separate entries in its output. Each of these was classified together with its respective preceding or succeeding coding sequence; failure to do so would make comparison practically impossible. GeneZilla likewise adds the additional genetic feature Poly(A) tail among its output; these were discarded as they are not included as part of the CDS annotations in GenBank.

DISCUSSION

The traditional genefinder measurement statistics are presented in Table 1. Focusing first at the exon level, GenomeScan seems the decisively best performing program with an average sensitivity and specificity of 0.75. Issac and Raghava confirmed this result in (21) where GenomeScan fared similarly with an average of 0.74. GlimmerHMM and Augustus converge on a similar performance level of 0.65. A step lower is Twinscan's exon average at 0.55. Finally Genezilla and SNAP (*Homo sapiens*) return an exon level average of 0.44 and 0.38, respectively.

At the nucleotide level Twinscan outperformed all the others with 0.88 and 0.89 for AC and CC, respectively. Twinscan's sensitivity and specificity measured 0.90 and 0.95, respectively. GlimmerHMM, GenomeScan and Augustus all returned AC and CC varying between 0.70 and 0.80. Finally SNAP outperformed Genezilla, yet both had an AC and CC value ranging from 0.65 to 0.69.

It is tempting to state that GenomeScan is the best performing genefinder overall; however, at the whole gene level it rated poorest, not finding 43 genes in the 294 (15%) sequences it successfully completed. Statistically this is worse than any other genefinder in previous independent evaluations (18,20). Previously Genie (22) and MZEF (23) were the worst performers not finding a gene in 7% of sequences tested. Genezilla and Augustus performed best in this area identifying a coding region in every sequence tested.

It is no surprise that the two lowest performing genefinders were those requiring partial or complete training, especially considering the overall lack of documentation and support in the genefinding software development world. Training for SNAP is mostly automated. Genezilla's complex training regimen however has a larger opportunity for human error. SNAP was designed to perform initial genefinding on sequences for which no organism-specific genefinder exists; furthermore, its state model is not designed specifically for higher eukaryotes. Genezilla, however, is a massive system implementing multiple specific sub-model types over a robust state structure.

A second explanation for the results of Genezilla and SNAP exists. Each genefinder returned similar results when trained on essentially the same dataset. Therefore, it is possible that the training dataset are responsible for their lower results.

Comparing our results to that of Rogic *et al.* (19), there does not seem to be a vast improvement in the genefinders tested. AC varied in (19) from 0.68 to 0.91. The six programs

Table 2. Distribution of predicted exons by feature

	Initial	Internal	Term	Single
Actual exon distribution	0.10	0.78	0.10	0.01
Mean for all genefinders	0.12	0.72	0.12	0.02
Augustus	0.11	0.73	0.13	0.03
Genezilla	0.11	0.77	0.11	0.01
GenomeScan	0.11	0.76	0.11	0.02
GlimmerHMM	0.12	0.73	0.13	0.02
Twinscan	0.08	0.70	0.05	0.01
SNAP (*H.sapiens*)	0.20	0.60	0.17	0.03

For each program the percentage of exons predicted as a particular feature is displayed. Term is an abbreviation for Terminal.

we tested produced an AC ranging from 0.67 to 0.88. The mean of exon specificity and sensitivity ranged from 0.43 to 0.76 in Rogic's tests, while most of the genefinders in this evaluation ranged from 0.44 to 0.75. Given these results and those at the whole-gene level discussed above, why have genefinders remained stagnant in their performance especially when they have individually published higher results? Have HMM genefinders reached their quantum limits? How can future HMM genefinder development proceed to be more effective in the future?

In order to answer these questions we have developed the PET. We submit that every predicted coding sequence must be placed into one of the 13 possible classes, and the patterns (or ratios) between classes considered in future genefinder development. Similar to the habit of identifying only four classes of coding regions, any classification of predicted exons that is not comprehensive provides inadequate information for properly ascertaining genefinder performance.

Future techniques must now focus on resolving these patterns of inaccuracy inherent in all HMM-based genefinders. Use of the PET and measuring the ratios between predicted exon classes will allow researchers to directly measure the effects of new genefinders.

FEATURE

Every sequence is classified by its genefinder as a particular feature, being initial, internal, terminal or single. Table 2 displays the average distribution of exons for all six genefinders. The top row of the table shows the actual distribution of exons. It can be seen that the genefinders achieve a high level of accuracy at 74% for internal exons, while the actual percentage of internal exons is 78%. For the remaining exon feature types each gene finder was within two percentage points of correctly identifying the appropriate number of exons. It is tempting to use this as an indicator of exon level performance; however, one must be careful because in some instances a genefinder will have predicted multiple start/stop exons on the same strand, while in others it has predicted no start/stop exons.

Predicted exon taxonomy

Looking at the results in Table 3, overall the genefinders found more class 1 exons, correct exons, than any other class. Eighty-two percent of the exons Twinscan found were class 1, while only 36% of SNAPs were class 1. On the average 63% of the exons predicted were correct; meaning that in general automated genefinders are right almost two-thirds of the time at the exon level. Agreeing with the EAVG results from Table 1, this makes the genefinders GlimmerHMM and Augustus average and GenomeScan above average.

The second and third largest classes of exons are 13 and 12, respectively, the wrong exons. The next most frequently occurring group of exons are those termed 'Partially Correct' (classes 3–6), where each exon correctly matches one boundary. Each class makes up ~2.8% of total number of predicted exons. For each of the overlapping exon classes 2, 7, 8 and 9, they each comprise ~0.5% of all exons predicted. Class 10 exons would break the standard splice site rules as a predicted exon has a 3′ boundary that matches a known 5′ donor site. Class 11 exons would have a 5′ boundary on an acceptor splice site. No exons of either of these last two classes were predicted.

A potential discrepancy appears between the class 1 exons of Table 3 and the CR percentage of Table 1. It would seem that these should be approximately equivalent as we are calculating statistics based on both exon boundaries matching. The difference however is in the method of calculating the average. The statistics in Table 1 are a normalized distribution (the sum of the correct exon frequencies for all sequences) of exons for each sequence. Conversely the averages in Table 3 have not been normalized, but are the simple count of predicted exons for each class divided by the total number of predicted exons.

Looking at the raw percentages of Table 3, it would seem that Twinscan is by far the most effective genefinder, almost ten percentage points above GenomeScan. However given its method of calculation, the high results for Twinscan may have been skewed by high accuracy on a few sequences with an abnormally high exon count. These results are not invalid, but indicate that Twinscan could be more effective on longer sequences. Further testing would be required to confirm this.

The purpose in creating this PET was to identify patterns of inaccuracy in all HMM genefinders. The following questions were posed: Is there any class of exon that is never predicted? If so how does this class relate to its boundaries? Do any patterns materialize around the 5′ or 3′ ends of a gene? Are there any patterns evident to 'internal' exons? Do intronless genes display any peculiar pattern? Which patterns occur in each class given a particular exon feature? How far away are incorrectly predicted boundaries from real boundaries?

In answering the first question no exons was correctly predicted in classes 10 and 11. Beyond this GenomeScan was the only genefinder that did not predict an exon of every class. It found no class 7 exons. The distribution of class 7 exons is displayed in Table 4. Again we see GenomeScan's continued effectiveness.

With regard to the 5′ boundary of a multi-exon gene it is clear (from Table 5) that class 5 initial exons tend to occur twice as much as class 3 initial exons. This shows that HMM genefinders are more conservative and tend to predict shorter initial exons on the 5′ side of the exon. It also shows that the 3′ end of the first coding exon tends to be accurately identified.

At the other end of the gene, class 4 terminal exons generally occur much more frequently than class 6 terminals.

Table 3. PET distribution

Class	Augustus (%)	GeneZilla (%)	GenomeScan (%)	GlimmerHMM (%)	Twinscan (%)	SNAP (%)	Avg (%)
1	72.19	49.00	73.28	65.33	82.42	36.08	63.05
13	11.54	31.67	14.41	18.09	4.73	32.75	18.86
12	5.24	5.15	4.31	5.74	1.37	9.63	5.24
5	2.49	5.15	1.73	2.09	2.61	4.58	3.11
4	4.27	2.00	2.04	2.70	0.83	5.79	2.94
3	3.63	2.67	2.29	3.25	2.20	3.26	2.89
6	1.06	2.26	1.36	1.26	4.48	3.50	2.32
8	0.21	0.68	0.19	0.38	0.25	2.12	0.64
2	0.38	0.29	0.19	0.51	0.17	1.31	0.47
9	0.38	0.64	0.42	0.44	0.41	0.44	0.46
7	0.13	0.48	0.00	0.21	0.54	0.54	0.32
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00

For each of the classes the mean average as a percentage of overall predicted exons was calculated. 5% of all exons predicted are class 12, while ~19% are class 13.

Table 4. Class seven exon distribution by genefinder and feature

Genefinder	Initial	Internal	Term	Single
Augustus	0	2	1	0
Genezilla	3	9	1	2
GenomeScan	0	0	0	0
GlimmerHMM	2	4	0	0
Twinscan	0	8	1	4
SNAP (*H.sapiens*)	6	9	1	0

GenomeScan was the only genefinder to predict zero exons for a class (no including classes 10 and 11 as no genefinder predicted these). Term is an abbreviation for Terminal.

Table 5. Initial and terminal exon comparison

	3 Initial	5 Initial	4 Terminal	6 Terminal	4 Single	6 Single
Augustus	12	22	26	1	6	0
GeneZilla	17	52	22	8	1	0
GenomeScan	2	15	5	3	1	0
GlimmerHMM	16	36	23	6	4	0
Twinscan	8	29	3	3	2	0
SNAP (*H.sapiens*)	35	74	92	3	8	0
SNAP (*A.thaliana*)	41	99	74	9	28	0

Instead of returning conservative results, genefinders are much more likely to predict exons extending into the 3′-UTR, than stopping before the stop codon. This is further supported by the fact that no genefinder predicted a class 6 single exon. Interestingly GeneZilla, whose output include Poly(A) tail annotation fared similarly to the other genefinders, when one might expect a genefinder with more states in its state model to be more accurate around the 3′ boundary.

Comparing the averages (Table 3) we see that class 13 exons occur more than three times as frequently as class 12. Further investigation showed that class 13 exons' average distance from their reference exon (5668 bases) is slightly more than half the distance of class 12 exons (10 489 bases). Thus genefinders seem more likely to predict exons nearer to the 3′ side of actual exons, and conversely are less likely to predict exons 5′ of a TE.

Before SNAP was trained on human DNA sequences, it was trained on *Arabidopsis*. All of patterns identified above, occurred to a similar degree in the SNAP (*A.thaliana*) results. First, class 5 initial exons were predicted twice as much as class 3 initial exons. Second, a tendency to predict exons which extend into the 3′-UTR region. Class 13 exons still occur ~3 times as often as class 12, and at an average distance (7064 bases) significantly closer to an actual exon than class 12 (12 164 bases).

Each of the patterns described above is a specific item for potential and measurable improvement in HMM genefinders. A new 5′-UTR-specific model may be ideal to reduce initial class 5 exons, while concurrently keeping class 3 low. The quality of new 3′-UTR and Poly(A) tail models can be assessed against previous results to ensure that the count of class 6 terminal exons is not increasing, if one reduces the number of class 4 terminal exons. Using the PET, one can compare models by class to ensure the count of exons of a particular class are decreasing (or increasing if it is class 1). One can measure the different ratios of each class to precisely verify the affects of a change to a model.

HOMOLOGY AND SIMILARITY

Two genefinders tested include homology by default, Genomescan and Twinscan. GenomeScan scored highest at exon level in Table 1, while Twinscan scored highest in class 1 exons in Table 3. The homologs used for Twinscan are incorporated into the system by the developer, whereas GenomeScan requires the researcher to provide homologous sequences. If a suitable homolog set is not available then neither of these two may provide high-quality results.

Furthermore regarding similarity, we can see SNAP's results on two different training sets, human and plant. The difference in results is clearly seen, with SNAP having much higher performance when trained on a human data for testing on human DNA sequences. This underscores an already prevalent theme that training on the same organism, if not at least homologous organism is vital for good genefinder performance.

CONCLUDING REMARKS

In summary we have evaluated the most recent HMM-based genefinders upon an independent test set, and it seems the

latest generation of genefinders does not perform vastly better than those tested 5 years ago. No one genefinder can be decisively named the best, but GenomeScan seemed to perform most noteworthy. The other homology incorporating genefinder, Twinscan produced solid results at the nucleotide level and in raw exons annotated correctly.

We also grouped every predicted exon into one of thirteen classes based on our comprehensive and novel taxonomy. With this we can more precisely measure the performance of all genefinders, and the effects a change has on their output. We identified four patterns of inaccuracy common to all HMM-based genefinders. As these patterns occur in all genefinders some fundamental shift may be needed to obtain consistently higher performance.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR online.

ACKNOWLEDGEMENTS

We would like to thank the two anonymous reviewers, Mikael Bodén, Justin Rough, Robert Dew, Jason Wells and Drew Quillam for their valuable time and insights. Funding to pay the Open Access publication charges for this article was provided by Australia Research Council Grant and Faculty of Science and Technology Deakin University.

Conflict of interest statement. None declared.

REFERENCES

1. Fickett,J.W. (1982) Recognition of protein coding regions in DNA sequences. *Nucleic Acids Res*., **10**, 5303–5318.
2. Meyer,I. and Durbin,R. (2002) Comparative *ab initio* prediction of gene structures using pair HMMs. *Bioinformatics*, **18**, 1309–1318.
3. Yeh,R.F., Lim,L.P. and Burge,C.B. (2001) Computational inference of homologous gene structures in the human genome. *Genome Res*., **11**, 803–816.
4. Hu,P. and Brent,M.R. (2003) Using TWINSCAN to predict gene structures in genomic DNA sequences. *Curr. Protocols Bioinformatics*, pp. 4.8.1–4.8.19.
5. Zhang,M. (2002) Computational prediction of eukaryotic protein-coding genes. *Nature Rev. Genet*., **3**, 698–709.
6. Mathé,C., Sagot,M.F., Schiex,T. and Rouzé,P. (2002) Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Res*., **30**, 4103–4117.
7. Wang,Z., Chen,Y. and Li,Y. (2004) A brief review of computational gene prediction methods. *Genome Proteomics Bioinformatics*, **2**, 216–221.
8. Rabiner,L. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**, 257–285.
9. Durbin,R., Eddy,S., Krogh,A. and Mitchison,G. (1998) *Biological Sequence Analysis Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge.
10. Yang,J., Xu,Y. and Chen,C. (1994) Gesture interface, modeling and learning. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE Computer Society Press, San Diego, CA, pp. 1747–1752.
11. Rabiner,L. and Juang,B.H. (1986) An introduction to hidden Markov models. *IEEE ASSP Mag*., **3**, 4–16.
12. Forney,J.D.,Jr (1973) The Viterbi algorithm. *Proc. IEEE*, **61**, 268–278.
13. Majoros,W.H., Pertea,M., Delcher,A.L. and Salzberg,S.L. (2005) Efficient decoding algorithms for generalized hidden Markov model gene finders. *BMC Bioinformatics*, **6**, 16.
14. Stanke,M. and Waack,S. (2003) Gene prediction with a hidden Markov model and a new intron submodel. *Bioinformatics*, **19**, ii215–ii225.
15. Majoros,W.H., Pertea,M. and Salzberg,S.L. (2004) TigrScan and GlimmerHMM: two open source *ab initio* eukaryotic gene-finders. *Bioinformatics*, **20**, 2878–2879.
16. Korf,I. (2004) Gene finding in novel genomes. *BMC Bioinformatics*, **5**, 59.
17. Korf,I., Flicek,P., Duan,D. and Brent,M.R. (2001) Integrating genomic homology into gene structure prediction. *Bioinformatics*, **17**, S140–S148.
18. Burset,M. and Guigó,R. (1996) Evaluation of gene structure prediction programs. *Genomics*, **34**, 353–367.
19. Rogic,S., Mackworth,A.K. and Ouellette,F.B.F. (2001) Evaluation of gene-finding programs on mammalian sequences. *Genome Res*., **11**, 817–832.
20. Altschul,S.F., Madden,T.L., Schäffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*., **25**, 3389–3402.
21. Issac,B. and Raghava,G.P.S. (2004) EGPred: prediction of eukaryotic genes using *ab initio* methods after combining with sequence similarity approaches. *Genome Res*., **14**, 1756–1766.
22. 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 States,D., Agarwal,P., Gaasterland,T., Hunter,L. and Smith,R. (eds), *Proceedings of the Fourth International Conference on Intelligent System for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 134–142.
23. Zhang,M.Q. (1997) Identification of protein coding regions in the human genome by quadratic discriminant analysis. *Proc. Natl Acad. Sci. USA*, **94**, 565–568.