BIOINFORMATICS ORIGINAL PAPER Vol. 28 no. 12 2012, pages 1571–1578
doi:10.1093/bioinformatics/bts176

Sequence analysis Advance Access publication April 18, 2012

Automated gene-model curation using global discriminative learning

Axel Bernal[1,*], Koby Crammer[2] and Fernando Pereira[1,3]

[1]Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104, USA, [2]Department of Electrical Engineering, Technion. Israel Institute of Technology, Haifa 32000, Israel and [3]Google Inc. Mountain View, CA, 94043, USA

Associate Editor: John Quackenbush

ABSTRACT

Motivation: Gene-model curation creates consensus gene models by combining multiple sources of protein-coding evidence that may be incomplete or inconsistent. To date, manual curation still produces the highest quality models. However, manual curation is too slow and costly to be completed even for the most important organisms. In recent years, machine-learned *ensemble* gene predictors have become a viable alternative to manual curation. Current approaches make use of signal and genomic region consistency among sources and some *voting scheme* to resolve conflicts in the evidence. As a further step in that direction, we have developed eCRAIG (*ensemble* CRAIG), an automated curation tool that combines multiple sources of evidence using global discriminative training. This allows efficient integration of different types of genomic evidence with complex statistical dependencies to maximize directly annotation accuracy. Our method goes beyond previous work in integrating novel non-linear annotation agreement features, as well as combinations of *intrinsic* features of the target sequence and *extrinsic* annotation features.

Results: We achieved significant improvements over the best ensemble predictors available for *Homo sapiens*, *Caenorhabditis elegans* and *Arabidopsis thaliana*. In particular, eCRAIG achieved a relative mean improvement of 5.1% over Jigsaw, the best published ensemble predictor in all our experiments.

Availability: The source code and datasets are both available at http://www.seas.upenn.edu/abernal/ecraig.tgz

Contact: abernal@seas.upenn.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on October 27, 2011; revised on March 8, 2012; accepted on April 3, 2012

1 INTRODUCTION

Ensemble predictors are automated gene-model curation tools that build gene models by combining conflicting and incomplete information provided by multiple sources of coding evidence. To date, manual curation still produces the highest-quality annotations as it benefits from problem-domain knowledge of expert human curators that is difficult to build into gene predictors. However, manual curation is slow and costly, and has been unable to complete the annotation even for the most important organisms (Baumgartner *et al.*, 2007).

Several methods of semi-automatic curation try to speed up manual methods by automating the selection of good combinations of manually-tuned evidence integration rules. Such methods include ExonHunter (Brejová *et al.*, 2005), Eugene (Schiex *et al.*, 2001), GAZE (Howe *et al.*, 2002) and more recently Augustus+hints (Stanke *et al.*, 2008). Semi-automated gene annotation pipelines also belong in this category. ENSEMBL (Potter *et al.*, 2004) and Pairagon+NSCAN_EST (Arumugam *et al.*, 2006) for example, use expressed sequence tags (ESTs), complementary DNAs (cDNAs) and automated predictions to find a consensus gene model.

In recent years, *ensemble* predictors based on machine learning have become a viable alternative to manual or semi-automated methods. These methods measure signal and genomic region consistency among sources encoded as feature vectors and use *voting schemes* to resolve evidence conflicts. Fully automated, learning-based *ensemble* predictors include the work of (Pavlović *et al.*, 2002) and JIGSAW (Allen and Salzberg, 2005). In particular, JIGSAW has performed favorably in multiple benchmark datasets in comparison with the best semi-automated annotation pipelines.

Some learning-based ensemble predictors require annotated sequences for training (*supervised* methods); others learn combination parameters to maximize agreement without needing annotated training data beyond what was used for individual sources of evidence (*unsupervised* methods). Unsupervised methods such as GLEAN (Elsik *et al.*, 2007) and Evigan (Liu *et al.*, 2008) are advantageous in cases where no manually-curated gene annotations are available. However, they tend to perform less well than supervised ensemble predictors.

Among supervised ensemble predictors, the work of (Pavlović *et al.*, 2002) uses maximum-likelihood estimation to learn the parameters of an input–output Hidden Markov Model (HMM) that combines multiple gene predictions. JIGSAW (Allen and Salzberg, 2005) is an improved version of Combiner (Allen *et al.*, 2004) that uses a semi-Markov structure model and a learning procedure that implicitly models dependencies of evidence sources given the input sequence. The main idea in Combiner and JIGSAW is to classify feature vectors of the input sources as either *accurate* or *inaccurate* using decision trees. An accurate (inaccurate) vector is one that led to predictions that match the annotation in at least (most) half of the training set. The learned decision trees partition the feature space into accurate and inaccurate regions, allowing the combiner to decide

*To whom correspondence should be addressed.

© The Author 2012. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oup.com

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

which predictions to trust where. This technique tries to emulate the way expert human curators decide on the structure of the final consensus gene model and it has proven to be quite effective in practice as Combiner and JIGSAW have consistently produced the best automated prediction results in a variety of benchmark datasets. However, this approach is unable to globally optimize the combiner to minimize error of its output consensus prediction.

In this article, we describe eCRAIG [*ensemble* CRAIG, since it is based on the earlier CRAIG gene predictor (Bernal *et al.*, 2007)], a learning-based automated curation tool that integrates multiple predictions through global *discriminative training*. This learning approach has been successfully used for *ab initio* gene prediction in CRAIG, which uses the margin infused relaxed algorithm (MIRA; Crammer *et al*., 2006) online learning algorithm to train a semi-Markov conditional random field (CRF). We use the adaptive regularization of weight vectors (AROW) learning algorithm (Crammer *et al.*, 2009), which combines the robustness to label-noise of large-margin classifiers like MIRA with the fast convergence and accuracy of confidence-weighted learning algorithm (Dredze *et al.*, 2008). The combined model is still a semi-Markov CRF, now using not just a rich variety of genomic features (as CRAIG does), but also features derived from the source predictions and their patterns of (dis)agreement. Training optimizes globally the feature weights to maximize annotation accuracy. The rich feature language allows the model to represent non-linear interactions involving *intrinsic* genomic properties of the sequence under analysis with *extrinsic* properties of the individual evidence sources as well as their patterns of agreement.

These refinements led to significant overall improvements over the current best predictions available for *Homo sapiens*, *Caenorhabditis elegans* and *Arabidopsis thaliana*. In particular, eCRAIG achieved a gene-level *relative mean improvement* (Stanke and Waack, 2003) of 5.1% over JIGSAW, the best published combiner-type predictor in all our experiments.

2 EXPERIMENTS AND RESULTS

We performed experiments on three different organisms: *A.thaliana*, *C.elegans* and *H.sapiens*. Table S-1 (see Supplementary Material) provides an overview of the training and test sets used in each case.

Predictions made by eCRAIG include partial genes, multiple genes per region and genes on both strands, but exclude alternatively spliced transcripts and genes within genes. Accuracy numbers for all programs were either reported from their corresponding publications if available, or computed using the program `eval` (Keibler and Brent, 2003).

As described in Section S-3 (see Supplementary Material), the eCRAIG learning algorithm requires the tuning of hyper-parameters r, N and loss function L. To accomplish this, for each experiment we constructed a development set—or validation set—using all *unreachable* genes found in the available training data. Unreachable genes are genes that have at least one genomic signal with no supporting evidence in all their transcripts. As such, they are inherently difficult to predict, impossible to decode correctly and thus could potentially cause overfitting if used for training. However, if used for validation, even a few unreachable genes could be enough to effectively measure prediction performance.

The number of unreachable genes in the original training sets for *A.thaliana*, *C.elegans* and *H.sapiens* are 21, 103 and 5, respectively. As shown in Figure S-1 (see Supplementary Material), the values for N, r and L that yield the best validation results are 4, 10 *and WS+MS* for *A.thaliana*; 5, 1.0 *and FP+FN* for *C.elegans*; and 6, 1.0 *and FP+FN* for *H.sapiens*.

For selecting control points for a particular property, we made sure that the two outermost control points covered 99% of the range for property values found in the training set; the remaining control points were placed equidistantly. For sparse property values, such as lengths, we used a log scale for the placements. The actual control point values are given in Tables S-9 and S-10.

2.1 Predictions for *A.thaliana*

We compared eCRAIG with *ab initio* predictors GeneMarkHMM, GenScan, GlimmerA, GlimmerM, TwinScan, and *ensemble* predictors GLEAN, Evigan and Combiner. We tested two different evidence sets for ensemble predictors. The first set integrates *ab initio* predictions with protein and EST alignment evidence, whereas the second one, with suffix '-NoAlign', only includes *ab initio* predictions. Figures 1a and b show gene- and exon-level prediction results for all programs on dataset ATTS1783. Accuracy values for all levels and types of exons are reported in Table S-2 (see Supplementary Material). Most ensemble predictors have an exon-level F-score of ∼92%, and a gene-level F-score of ∼80%. These results are significantly better than those obtained by the best *ab initio* predictor (Twinscan), showing the benefit of evidence combination. Furthermore, eCRAIG achieves the highest overall accuracy at all levels both with and without alignment evidence. In both cases, specificity improves slightly more than sensitivity across all levels, supporting the hypothesis that global discriminative training is advantageous in evidence combination.

Overall, eCRAIG performs the best among of all methods, except for a somewhat lower single exon sensitivity but higher single exon specificity than eCRAIG-NoAlign. The absolute F-score improvement of eCRAIG over Combiner, the second-best program overall along with eCRAIG-NoAlign (see Fig. 1a and b), is 5.8% for complete gene structures. The absolute F-score improvement at the gene level for eCRAIG-NoAlign over Combiner-NoAlign is 3.8%. eCRAIG improves over eCRAIG-NoAlign on internal and terminal exons with an absolute F-score of 1.7% and 2.2%, respectively. Single and initial exon predictions also improve somewhat, which is expected as the alignments are of better quality near splice signals than translational signals. The improved gene-level accuracy follows from these gains at the exon level. Evigan(-NoAlign) performs worse than Combiner(-NoAlign) in most categories whereas *ab initio* gene finders perform much worse than ensemble predictors at all levels.

2.2 Predictions for nGASP data

The nGASP project (Coghlan *et al.*, 2008) is a *C.elegans* gene annotation initiative which emulates the efforts made by EGASP (Guigo and Reese, 2006) for *H.sapiens*. Several groups involved in computational gene prediction research were invited to submit gene predictions on 10 non-overlapping 1 Mb regions of *C.elegans* genome assembly build WS160. Ten regions of similar size were also provided as training data. The organizers asked the participating groups to train new gene models using only the given training set. The nGASP evaluation had two rounds. In the first round, the contest was open to all Category 1–3 predictors: *ab initio*,

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

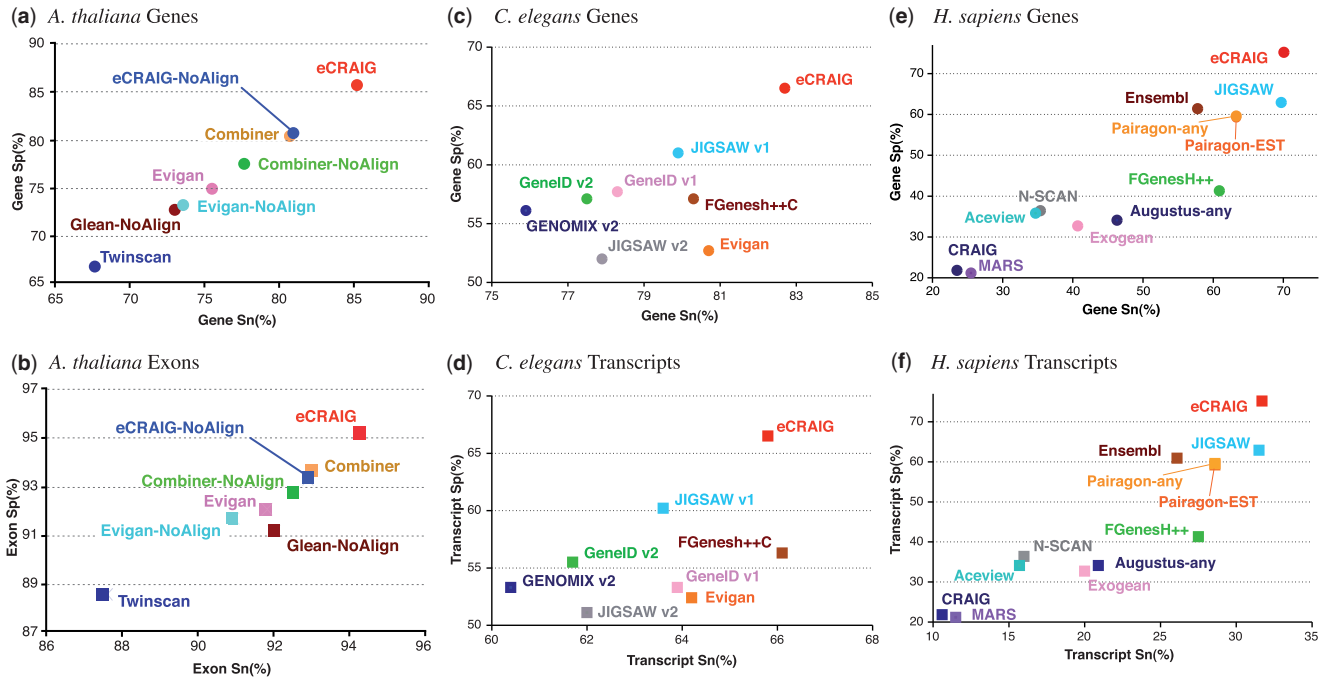

Fig. 1. Gene and transcript/exon-level accuracy results for all relevant predictions evaluated for each experiment. Accuracy results are given as Sn versus Sp graphical plots for *A.thaliana* (**a,b**), *C.elegans*' NGASP (**c,d**) and *H.sapiens*' ENCODE (**e,f**) datasets. Each subfigure uses a different scale for better resolution as results vary greatly among experiments

de novo and reference-based. In this phase, alignments to closely related organisms and EST and protein data for all regions were made readily available as auxiliary information. The second round, open to ensemble-type predictors, allowed predictors to use all round-one predictions as well as the original auxiliary information. Sensitivity and specificity were evaluated with eGASP metrics.

Seventeen groups participated and submitted 44 prediction sets to nGASP; out of these, 12 sets corresponding to Category 4 predictions were filtered out. Some groups submitted multiple prediction sets using different running parameters; out of these, we removed redundant sets: Category 3 predictors MGENE v2, EUGENE v1 and MAKER+SNAP v2; Category 2 predictor MGENE v2; and Category 1 predictors AUGUSTUS v1, MGENE v1 and SNAP. Finally, we did not include Category 1 predictors Agene and ExonHunter due to their very poor prediction performance and to avoid the extra computational cost. The final set of evidence sources which were used as input for training eCRAIG are listed in Table S-3 (see Supplementary Material).

The current version of eCRAIG can only accomodate evidence for a single transcript at each gene locus. This posed a problem when incorporating prediction sets containing multiple transcripts at a single locus. In these cases, we removed all transcripts but the longest one; the actual number of transcripts used as input evidence for eCRAIG is shown in Table S-3 (see Supplementary Material).

Following the nGASP guidelines for performance evaluation, we added a transcript-level prediction category in our reports to better evaluate predictions on alternatively spliced genes. Accuracy results at the gene and transcript level for all these submissions as well as eCRAIG's are presented in Figures 1c and d. Actual prediction accuracy values for all levels can be found in Table S-4 (see Supplementary Material). Performance numbers for predictors other than eCRAIG were obtained from Coghlan *et al.* (2008). Evaluation of eCRAIG's predictions followed the same procedure established by the nGASP organizers to assess prediction performance, that is to say, sensitivity (Sn) at all levels was obtained using only full-length cDNAs as reference (set NGASPA313) whereas specificity (Sp) was obtained using all manually curated genes in the test regions, both experimentally confirmed and unconfirmed (set NGASPC966).

As shown in Figure 1c and d, eCRAIG outperforms all other programs at the gene and transcript level. The performance gap between eCRAIG and the second-best program, JIGSAW v1, measured by absolute F-score improvement, is 4.4% at the transcript and 4.7% at the gene level. These improvements are similar to those in our *A.thaliana* experiment above: scores are improved overall, with specificity gains being slightly higher than sensitivity gains. Some programs do well at predicting individual exons, but less well at assembling exons into correct transcripts, especially in the presence of alternative splicing. As a result, eCRAIG's gains relative to those programs are lower at the exon level than at the transcript level.

Programs capable or predicting multiple transcripts at each gene locus reported fewer number of false negative exons at the cost of a much higher number of false positives. For example, EUGENE cat4 v2 predicts 1638 transcripts—almost 70% more than the annotation—and finds 32 more true positive transcripts than eCRAIG. However, EUGENE cat4 v2 also predicted 301 more false positive transcripts than eCRAIG.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

2.3 Predictions for the ENCODE regions

Our final set of experiments are on *H.sapiens* genomic data. The test set ENCODETS294 consists of 31 regions from the ENCODE project (ENCODE Project Consortium, 2004; Guigo and Reese, 2006), for a total of 21 M bases, containing 294 carefully annotated alternatively spliced genes. After eliminating non-coding transcripts, the total number of transcripts remaining for this set is 650. We used the EGASP reference set of genes from October 2005 [(build hg17) (Guigo and Reese, 2006)] as our gold standard.

To handle prediction sets with multiple transcripts per locus, we proceeded in similar way to the nGASP experiment above, retaining only the longest transcript for each locus. This procedure affected prediction sets corresponding to predictors ENSEMBL, Aceview, MARS, Exogean and FgenesH++.

As with *C.elegans*, we added a transcript-level prediction category in our evaluation and closely followed the evaluation procedures proposed by (Guigo and Reese, 2006) and (Coghlan *et al.*, 2008). Following nGASP guidelines for training ensemble predictors, we trained eCRAIG on set ENCODETR137 but using only annotations from input sources which are not combiners themselves. From the 16 available sources, we used only the top nine as input, ranked by their accuracy on the training set. These sources are ENSEMBL, Pairagon+mRNA_EST, NSCAN, Aceview, Exogean, ExonHunter, MARS, Twinscan and CRAIG. Including other sources (see Section 4) did not increase prediction accuracy and slowed down training and decoding considerably.

The absolute F-score improvements at the transcript and gene levels are 3.2% and 7.3% over JIGSAW (see Figs 1e and f), yet again the runner-up program. Actual accuracy values for all levels are shown in Table S-5 (see Supplementary Material).

As in previous experiments, eCRAIG achieves higher gains in specificity than in sensitivity at all levels. At the transcript and gene level, eCRAIG predicts with slightly more sensitivity and considerably more specificity than JIGSAW. At the exon level, eCRAIG predicts correctly ∼5% fewer exons than JIGSAW, but it also makes ∼5% fewer mistakes. However, eCRAIG predicts 15% more correct gene structures and 11% fewer TIS false positives than JIGSAW. Furthermore, there are 75 genes that eCRAIG predicts correctly but JIGSAW either misses or predicts wrongly averaging 9.8 exons and 56 kb in length; conversely, JIGSAW predicts correctly 43 genes averaging 6.3 exon and 23 kb in length that eCRAIG either misses of predicts incorrectly. These results imply that eCRAIG predicts *longer* transcripts—typically those with a higher number of exons—significantly better than JIGSAW as the latter would sometimes miss a long transcript in favor of two shorter, often incorrect ones.

2.4 Significance testing

We tested the statistical significance of eCRAIG's improvements at the transcript level by comparing its results against the runner-up program for each experiment. Any transcript belonging to a particular test set is associated with two dependent Bernoulli random variables that indicate whether it was correctly predicted by eCRAIG and by the runner-up program. We compute *two-tail* p-value upper bounds with McNemar's test for dependent, paired samples extracted from predictions made by eCRAIG and the runner-up in each case. The null hypothesis is that eCRAIG's advantage in transcript predictions is due to chance. The p-values, shown in Table S-6 (see Supplementary Material) are <0.05 for all the experiments, which implies that eCRAIG's improvements are unlikely to be due to chance.

3 METHODS

3.1 Background

In what follows, $x=x_1\cdots x_P$, $x_i\in\Sigma_{\mathrm{DNA}}=\{\mathtt{A},\mathtt{T},\mathtt{C},\mathtt{G}\}$, is the input or target sequence, $\boldsymbol{s}=s_1\cdots s_Q$ is a *segmentation* of x, where each *segment* $s_j=\langle p_j,l_j,y_j\rangle$ starts at position $\mathrm{pos}(s_j)=p_j$, has length $\mathrm{len}(s_j)=l_j$ and state label $\mathrm{lab}(s_j)=y_j$, where $p_{j+1}=p_j+l_j\leq P$, $1\leq l_j\leq B$ for some upper bound B and $\mathrm{lab}(s_j)$ is a state in the finite state machine (FSM) that models the gene structure shown in Figure S-2 (see Supplementary Material).

Given $\mathcal{E}$, the set of evidence sources, we define $x_e=x_{e,1}\cdots x_{e,P}$, $x_{e,i}\in\Sigma_{\mathrm{EV}}=\{\mathtt{i},\_,\mathtt{E},.\}$, the *evidence sequence* associated to source $e\in\mathcal{E}$, as follows:

$$x_{e,i}=\begin{cases}\mathtt{i} & \text{if } \exists s\in\boldsymbol{s}_e|i\in\mathrm{range}(s) \text{ and} \mathrm{lab}(s)=\mathrm{I}_p,\ p=0,1,2\\ \mathtt{E} & \text{if } \exists s\in\boldsymbol{s}_e|i\in\mathrm{range}(s) \text{ and} \mathrm{lab}(s)=\mathrm{E}_{\mathrm{frame}(i,s)}\\ . & \text{if } \exists s\in\boldsymbol{s}_e|i\in\mathrm{range}(s) \text{ and} \mathrm{lab}(s)=\mathrm{IG}\\ \_ & \text{otherwise}\end{cases}$$

where source e is encoded as a segmentation $\boldsymbol{s}_e$ over x, $1\leq i\leq P$, $\mathrm{range}(s)=[\mathrm{pos}(s),\mathrm{pos}(s)+\mathrm{len}(s)-1]$, p is the segment phase and $\mathrm{frame}(i,s)=(i-\mathrm{pos}(s)) \mod 3$. For *H.sapiens* and *C.elegans*, evidence sequences distinguish between short (`i`) and long introns (`l`). Figure 2 shows this notation applied to an example in *A.thaliana*.

Linear structure models for *ab initio* gene prediction were first introduced in (Bernal *et al.*, 2007, pp. 494–495). In this article, we apply these models to the *ensemble* gene prediction problem with two notable extensions. First, online parameter updates are now computed using AROW instead of MIRA and second, our learning procedure can now train with alternatively spliced instances. Detailed descriptions of these extended models are given in Section S-1 (see Supplementary Material).

For gene prediction, we need to define three basic components. First, an *inference* method that can efficiently find the best-scoring segmentation for a given sequence x. Second, a *training* method that can learn weights using dataset $\mathcal{T}=\{(x^{(t)},\boldsymbol{s}^{(t)})\}_{t=1}^{T}$, such that the best-scoring segmentation of $x^{(t)}$ is close to $\boldsymbol{s}^{(t)}$. Finally, we need to select a feature function $\boldsymbol{f}$ that is compatible with the first two components while providing good generalization to unseen test sequences. The inference and online learning algorithms are described in Sections S-2 and S-3 (see Supplementary Material), respectively. Feature function $\boldsymbol{f}$ is described in the next section.

Fig. 2. Target sequence x, annotated segmentation $\boldsymbol{s}$ and aligned evidence sequences x_e, $e\in\{e_1,e_2,e_3,e_4,e_5\}$ for a fragment of *A.thaliana*'s gene annotation_60464.08. We use $\mathrm{E}_p(\mathrm{I}_p)$ to represent all types of exons(introns) in phase p. These phases are computed as the CDS length (mod 3); the CDS length is measured up to the end of the previous segment. For instance, s_3 represents an intron in phase 16 (mod 3)$=1$. Phases are reset to zero for intergenic regions. Notice how, at testing time, coding exon phases could be guessed by examining x_e's triplet occurrences. For example, segment s_4 is likely an exon in phase 1 since the triplet following the acceptor is of the form `-E` in four out of the five available sources of evidence

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

3.2 Features

The main component of our model is a feature function $\boldsymbol{f}$ used to score candidate segments based on properties of the input sequence. A typical feature relates a proposed segment to some property, i.e. a real-valued function of the input sequence around that segment, and possibly to the label of the previous segment.

We distinguish between state and transition features. The former represent the content properties of a given *genomic region* whereas the latter look at biological signals and indicate a switch in *genomic region* type. Features testing for those signals look for motif enrichment within a window centered at a given offset from the position where the region switch occurs.

In all our experiments, feature functions do not depend on the strand of the genomic signal or region given as argument, so we only give the feature descriptions for the forward strand. Features looking at genomic regions or signals in the reverse complementary strand receive as input the reverse complemented sequence.

First, we introduce a variation of *binning* (Bernal *et al.*, 2007, p. 496), also described in Section S-4.2 (see Supplementary Material), to handle multimodal and/or sparse property value distributions.

3.2.1 Piecewise-linear binning We found that a better alternative to *binning* is to pass property values through a continuous piecewise-linear function instead of a piecewise-constant one. To this end, we initially proceed as in *binning*, i.e. we split the range of property values into disjoint intervals and their associated tests. In contrast to *binning*, each interval test is now associated with two real-valued functions instead of a boolean function. These real-valued functions are associated to the interval's left and right boundaries and their values add to one.

More formally, let $\boldsymbol{c}=c_1,\ldots,c_m$ be the interval boundaries, also called *control points*, let v be the property value and let $[c_l,c_r[$ be the interval selected for property value v, where:

$$\langle l,r\rangle=\begin{cases}\langle i,i+1\rangle & c_i\le v<c_{i+1}\\ \langle 1,1\rangle & v<c_1\\ \langle m,m\rangle & c_m\le v\end{cases}$$

The functions associated with the control points $c_l\le v<c_r$ are $1-h(v,\boldsymbol{c})$ and $h(v,\boldsymbol{c})$, respectively, where

$$h(v,\boldsymbol{c})=\begin{cases}\frac{v-c_i}{c_{i+1}-c_i} & c_i\le v<c_{i+1}\\ 1 & \text{otherwise}\end{cases}$$

Using this procedure, the property value v is transformed into a linear interpolation of the *heights* of the interval boundaries c_l and c_r. These heights are learned as weights of the real-valued feature functions associated with the interval's left and right boundaries. For regular w-wide intervals, we denote this transformation by $\text{Lbins}_w(v)$.

3.2.2 Piecewise-N-linear binning Piecewise-linear binning can be generalized to N-dimensional vectors of correlated real-valued properties. Here, control points defined for each property are placed along each dimension with the goal of constructing a N-dimensional grid where the *height* of each grid point is learned as the weight of its associated feature.

Given property values $v_1,\ldots,v_N$ and control point sets $\boldsymbol{c}^1\cdots\boldsymbol{c}^N$, it is easy to see that the set of intervals $[c^i_{l_i},c^i_{r_i}],\ 1\le i\le N$ selected for each value v_i form a hyper-rectangle. The set of features associated to each corner of this hyper-rectangle, denoted by S_N, is given by the following recursive formula:

$$\begin{aligned}\mathrm{S}_1&=\{1-h(v_1,\boldsymbol{c}^1),h(v_1,\boldsymbol{c}^1)\}\\ \mathrm{S}_i&=\left\{(1-h(v_i,\boldsymbol{c}^i))\times f,\ h(v_i,\boldsymbol{c}^i)\times f\setminus f\in\mathrm{S}_{i-1}\right\}\end{aligned}\qquad(1)$$

It is easy to see that property values $v_1\cdots v_N$ are transformed into a N-linear interpolation of the *heights* at each corner of the hyper-rectangle formed by the selected intervals $[c^1_{l_1},c^1_{r_1}],\cdots,[c^N_{l_N},c^N_{r_N}]$. For regular w-wide intervals used across all property values, we denote this transformation by $\text{N-Lbins}_w(\{v_1,\cdots,v_N\})$.

Equation (1) shows the complexity in the number of features to be exponential in N, the number of dimensions. This is a theoretical limitation of our model.

3.3 State features

3.3.1 Agreement features: These features measure how much different sources of evidence agree on the analysis of the target sequence. To measure agreement, we start by summarizing each evidence sequence x_e, $e\in\mathcal{E}$ with a set of n-gram counts. The simplest agreement features count 3-grams u in phase $q=0,1,2$ on evidence sequence x_e and are denoted by $\text{count}_{e,u,q}$. We also define agreement features $\text{dcount}_{e,u,u',q}$ that summarize the distribution of $\text{count}_{e,u,q}$ counts for each 3-gram u' in target sequence x.

Counting n-grams for each source separately can suffer from data sparsity, so we also use aggregate features that simply count the number of sources that agree on a particular n-gram and phase, $\text{ovcount}_{u,q}$. Count features for individual sources can also be conjoined to extract more fine-grained features from source agreements. Conjunctions between S $\text{count}_{e,u,q}$ features are denoted with $\text{countcnj}^{e_1,\cdots,e_S}_{u_1,\cdots,u_S,q}$.

All the agreement features mentioned above have a predictive effect that is a linear function of their value. However, typically, evidence sources display a varying predictive effect that depends on their level of agreement with the reference annotation. This non-linear effect can be achieved by splitting the range of agreement counts into *bins*, as described in S-4.2 (see Supplementary Material). We refer to this function as agreement count duration, since it constructs a explicit *length* model for agreement count distributions. To handle sparse observations in the tail region of these distributions, we take the $\log_2$ of the original feature values as input. We only define duration models for the $\log_2$-scaled values of features $\text{count}_{e,u,q}$, $\forall u,\ u=1$ and $\text{countcnj}^{e_1,e_2}_{u_1,u_2,q}$, $\forall u_1,u_2,\ |u_1|=|u_2|=1$. These restrictions are necessary due to data sparsity and computational complexity problems that arise for large values of u.

None of the agreement features mentioned above are defined for intergenic states. A formal, detailed description of these features is given in Section S-5 (see Supplementary Material).

Figure 3 shows how most of these agreement features are extracted from four sources and a reference annotation in *A.thaliana*.

3.3.2 Coding quality score: If an evidence source e provides a quality measure of its annotation, we define $\boldsymbol{q}_e=q_{e,1}\cdots q_{e,P}$ as the alignment quality scores associated with source e. These scores are zero for positions not in phase with a coding region. The coding quality score is computed in the following way:

$$\text{codQ}_e(s,x)=\frac{1}{\text{len}(s)}\sum_{k=\text{pos}(s)}^{\text{pos}(s)+\text{len}(s)}q_{e,k}$$

This sum is computed over non-overlapping codon scores instead of base scores to capture coding phase information.

Coding quality scores could also display a non-linear predictive effect. To model this behavior, we use piecewise-linear binning over the range of codQ_e values, as described in Section 3.2.1. This method performed significantly better than simple binning in all experiments.

3.3.3 Coding quality score correlations: Let $\mathcal{E}'$ be the set of evidence sources with associated quality scores. To capture correlations of coding quality scores among multiple evidence sources e, $e\in\mathcal{E}'$, we use piecewise-n-linear binning, as described in Section 3.2.2. Here the binning is over the range of feature values codQ_e, $e\in\mathcal{E}'$. As mentioned before, this multidimensional binning generates a number of features that is exponential in $|\mathcal{E}'|$. However, $|\mathcal{E}'|$ is typically a small value in practice—it is 2 in our experiments. Furthermore, correlations can be computed as pairwise relations or precomputed and cached if needed, so the computational cost was not high in practice.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

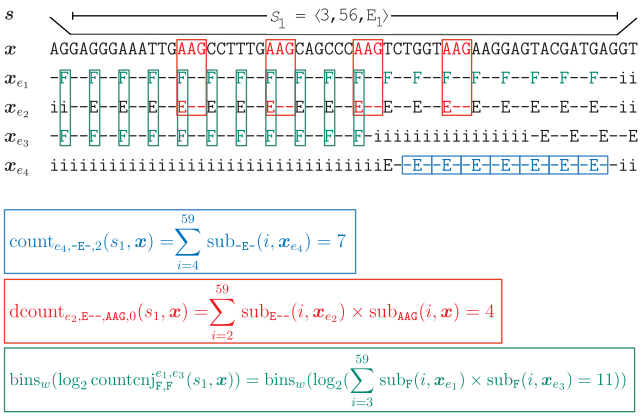

Fig. 3. Agreement features extracted from *A.thaliana* with four evidence sources. Features were computed for the second exon of gene annotation_60464.08. Sources e_1 and e_3 use the extended alphabet $\Sigma_{EV'}$ to generate x_{e_1} and x_{e_3}, respectively. The tests $\text{sub}_u(i,x)=[[u=x[i:i+|u|-1]]]$ check whether substring u occurs at position i in x. The colored text in each sequence shows where the corresponding $\text{sub}_u(i,x)$ is true. The colored frame boxes are counted toward the running total value of the associated feature. The index i ranges over codons for features $\text{count}_{e_4,\texttt{-E-},2}$ and $\text{dcount}_{e_2,\texttt{E--},\texttt{AAG},0}$, and over nucleotides for phaseless features like $\text{countcnj}_{\text{F,F}}^{e_1,e_3}$. Feature bins_w denotes a w-wide interval piecewise constant binning applied over the range of $\log_2 \text{countcnj}_{F,F,q}^{e_1,e_3}(s_1,x)$ values. Agreement features are described in detail in Section S-5 (see Supplementary Material)

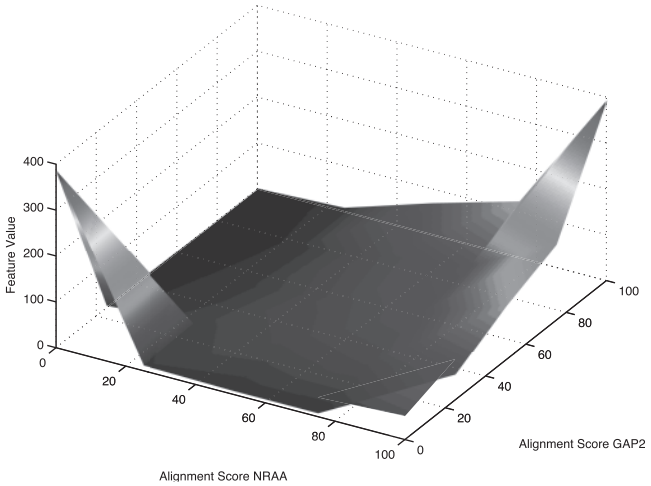

Fig. 4. Piecewise-2-linear binning feature values for NRAA (protein alignment) and GAP2 (EST alignment) scores for *A.thaliana*. Feature values were computed for coding regions in the entire training dataset. The alignment quality scores range from 0 to 100. A peak in the surface indicates strong agreement between the sources. For example, the peak at coordinates $(0,0)$ and $(100,100)$ in the XY plane, indicate that a significant portion of coding regions were either not aligned or aligned with the highest score in both sources simultaneously. The feature values computed at each corner of any rectangle of this two-dimensional grid are: $(1-h(v_1,\boldsymbol{c}^1))(1-h(v_2,\boldsymbol{c}^2))$, $(1-h(v_1,\boldsymbol{c}^1))h(v_2,\boldsymbol{c}^2)$, $h(v_1,\boldsymbol{c}^1)(1-h(v_2,\boldsymbol{c}^2))$ and $h(v_1,\boldsymbol{c}^1)h(v_2,\boldsymbol{c}^2)$, where function h is described in the text

Figure 4 displays the feature values computed for all the training dataset of a piecewise-2-linear binning of codQ_e quality scores provided by protein alignment source NRAA and EST alignment source GAP2 in *A.thaliana*.

Table S-9 (see Supplementary Material) shows a summary of all the state features associated with each genomic region type.

3.4 Transition features

3.4.1 Motif features: Motif features look for combinations of particular motifs within a window at a given offset from a state transition. These features, denoted by $\text{motif}_{e,o,u,l,p}$, where o is the offset, u is the enriched motif, l is the window width and $p=0,1,2$, test for motif occurrences around potential genomic signal locations in the target sequence. These locations correspond to positions annotated as genomic signal occurrences by some evidence source $e\in\mathcal{E}$. This approach gave better accuracy and decoding efficiency than using the signal consensus sequences typical in *ab initio* gene prediction.

It is straightforward to define windowed weight array models (WWAMs; Burge and Karlin, 1998) as feature sets $\text{WWAM}_{e,n,l,q,r,p}(s,y',x)=\{\text{motif}_{e,o,u,l,p}(s,y',x_e): u\in\Sigma_{\text{EV}}^n, q\le o\le r\}$, where q and r are set so that the WWAM can capture both motif occurrence and phase information from each source. A position weight matrix (PWM) is a special case of WWAM and can be defined as $\text{PWM}_{e,q,r,p}=\text{WWAM}_{e,1,1,q,r,p}$.

In situations where data sparsity is a problem, features that simply count the number of sources agreeing on a motif with the reference annotation could be useful. We call these features overall motifs and denote them by $\text{ovmotif}_{o,u,l,p}$. These counts are piecewise-linearly binned to incorporate a non-linear predictive effect on the number of sources agreeing on motif u.

A visual representation of some of these WWAM-based features for a region in *A.thaliana* is given in Figure 5. For a more formal, detailed description of these features refer to Section S-6 (see Supplementary Material).

3.4.2 Signal quality score: Quality scores can also be associated with signal occurrences. Let $\mathcal{E}'$ be the set of evidence sources that are either

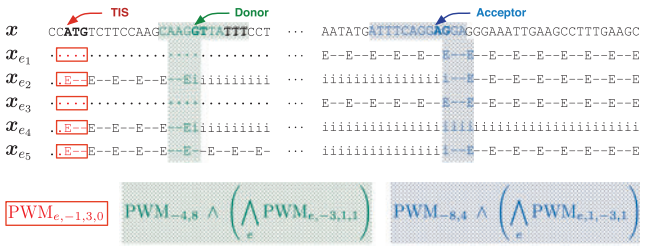

Fig. 5. Transition features extracted from *A.thaliana* with five evidence sources. Features values were computed for the first and second exons of gene annotation_60464.08 and for all sources. The colored text in each sequence is within the boundaries of the appropriate transition feature. Colored frame boxes represent simple PWMs, whereas boxes filled with dot patterns represent PWMs that model position-specific correlations through feature conjunctions as described in the text

alignments or provide quality measure scores. For $e\in\mathcal{E}'$, we define $\boldsymbol{q}'_e=q'_{e,1}\cdots q'_{e,P}$ as the corresponding signal quality score. These scores will be zero for positions where e does not indicate a signal.

We proceed similarly as in coding quality scoring features and use piecewise-linear binning over the range of $q'_{e,i}$ values, $1\le i\le P$.

3.4.3 Signal quality score correlations: To capture correlations of signal quality scores among multiple evidence sources e, $e\in\mathcal{E}'$, we use piecewise-n-linear binning, as described in Section 3.2.2. The setting is similar to the one applied for coding quality score correlations, but here the binning is over the range of $q'_{e,i}$, $e\in\mathcal{E}'$, $1\le i\le P$ values.

Table S-10 (see Supplementary Material) shows the summary of transition features defined for each biological signal.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

Table 1. Gene-level (Sp) and (Sn) for eCRAIG and its variants for each of the experiments considered here

	A.thaliana			*C.elegans*		*H.sapiens*	
	Base	NoLBin	NoDConj	Base	NoDConj	Base	NoDConj
Sn	85.0	82.5	83.6	82.7	81.5	70.8	61.9
Sp	85.7	84.1	84.0	66.8	64.5	76.2	64.1

Descriptions of eCRAIG-NoDConj and eCRAIG-NoLBin are given in the text. Note, eCRAIG-NoLBin was only trained and tested on the *A.thaliana* experiments, as those were the only ones including alignment quality scores as input evidence.
Sp, specificity; Sn, sensitivity

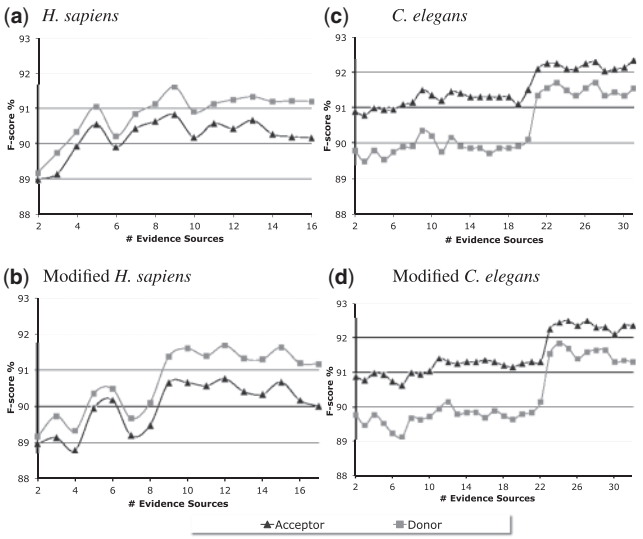

Fig. 6. Splice site prediction accuracy versus number of supporting evidence sources. Acceptor and donor prediction accuracy in (**a**) *H.sapiens* EGASP and (**c**) *C.elegans* nGASP test datasets. The *x*-axis represents the number of evidence sources for the combined model. Experiments (**b**) and (**d**) have a source added two times consecutively. For the *H.sapiens* modified experiment in (b), we added N-SCAN predictions at iterations 3 and 4; for the *C.elegans* experiment (d), we added predictions from Category 2 gene finder Eugene v1 at iterations 5 and 6. In both cases, splice site accuracy dropped relative to experiments (a) and (c). However, when enough sources were included—9 different sources in case of human and 23 in case of *C.elegans*—accuracy reached roughly the same levels in both experiments

4 DISCUSSION

4.1 Unreachable genes

Our prediction results did not account for the presence of unreachable genes in the test datasets; as such, all our results are biased toward lower sensitivity values at all levels. The statistics for unreachable genes and signals as well as the sensitivity of *oracle* predictions at the transcript, gene and signal levels are given in Table S-11 (see Supplementary Material). For example, the highest possible transcript-level prediction sensitivity that could be achieved in the EGASP test dataset is ∼63%; at 32.0% sensitivity, eCRAIG predicts slightly more than half of the EGASP transcripts correctly. On the other hand, at a gene-level sensitivity of 70.8%, eCRAIG missed or predicted incorrectly only 34 out of the 241 reachable EGASP genes, or 14.1%.

4.2 Accuracy effect of complex features

Our gene models integrate features such as durations of agreement count conjunctions and quality score multidimensional correlations, that are either conceptually or computationally complex. Here, we investigate whether the predictive effect of these features is worth their inclusion. For this, we trained and tested '-NoDConj', to denote a variant of eCRAIG that removes all duration features for agreement count conjunctions. For *A.thaliana* only, we used '-NoLBin', to denote a variant of eCRAIG that uses simple binning and binning conjunctions instead of linear (and *n*-linear) binning for modeling both state and signal quality scores and their correlations. The results at the gene level for these variants compared with the base model for each experiment are shown in Table 1.

The results are as expected: *n*-linear binning features and durations of agreement count conjunctions improve accuracy across all experiments. In particular, the predictive effect of duration conjunction features on EGASP improved gene- and transcript-level accuracy by ∼10% in absolute *F*-score. An effect of this magnitude was not observed in *A.thaliana* or *C.elegans*, presumably because potential gains might have been offset by having sufficient training data. Detailed results of these comparisons and an analysis of the relevance of other features for prediction are given in Table S-12 and Section S-7 (see Supplementary Material).

4.3 Accuracy effect of the evidence source set

It would be expected for prediction accuracy to increase as more external annotations are integrated as evidence sources, provided the features extracted from these sources are conditionally independent given the input sequence. However, conditional independence is not guaranteed, as sources typically are created using related algorithms, or even the same algorithm ran with different parameters. To quantify the accuracy effects as more evidence sources are added, we conducted a series of experiments using an increasing number of non-redundant evidence sources ordered by their agreement level with respect to the annotation. Sources deemed redundant (see Section 2) were added last. In a second round of experiments, we added a single source two consecutive times at some point early in the series to illustrate the negative effects of feature dependencies and whether the model recovers from its inclusion. At each point, we computed splice site prediction accuracy on the original test set and plotted these results for organisms *H.sapiens* and *C.elegans* as shown in Figure 6.

4.4 Global discriminative learning

Our experiment results have shown that global discriminative learning can achieve superior results in ensemble-based gene prediction. A similar conclusion had been reached earlier for the simpler problem of *ab initio* gene prediction (Bernal *et al.*, 2007). Yet, our present work uses a refined learning algorithm better able to cope with the wide range of features and feature distributions in genomic data.

Discriminative models involve fewer conditional independence assumptions on observations and labels than generative models such as HMMs, thus allowing us to combine a wide range of

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

weakly informative features to learn models that directly maximize annotation accuracy. We have benefitted from those advantages in designing the rich, varied features discussed in Section 4.2, and especially in using conjunctions to model feature interactions as seen in Section 4.3.

However, discriminative models are also more prone to overfitting due to insufficient training data, and discriminative learning algorithms for complex problems like gene prediction need to be carefully tuned to actually get close to the theoretical optimum of the learning objective. In particular, the training algorithm used in our CRAIG *ab initio* predictor, MIRA/PA, does not converge well when features have very different dynamic ranges and distributions in the training data, leading us to adopt the newer AROW algorithm. Accuracy results obtained by MIRA-trained models are shown in Figure S-3 and Table S-12 (see Supplementary Materials). These results are comparable to JIGSAW but significantly lower than eCRAIG. This improved performance can be explained by AROW's ability to keep track of weight confidence information during learning. Intrinsic attributes of gene structures, such as variable state lengths, bring about features that scale differently, appear in different fractions of the input data and have different scopes. For instance, counting features that look at sequence positions will occur much more often than features that look at entire segments, such as lengths. For features that are rare or *sparse*, methods like MIRA and perceptron do not attribute enough weight during training because such features are also updated rarely. However, when using confidence information, the weights for rare features have lower confidence and are thus updated more aggressively than the higher-confidence weights for features seen often in training. In other words, AROW, in contrast to MIRA and perceptron, can distinguish between features that are important and features that are reliable. This ability of AROW to better handle data sparsity and rare features made it also possible for eCRAIG to successfully learn from relatively small training sets. For instance, we only used 129 genes and 8 Mb of sequence data to learn our *H.sapiens* model, compared with 1500 genes which were reportedly used for training JIGSAW.

ACKNOWLEDGEMENTS

We thank Alex Kulesza and Mark Dredze for their advice on the implementation of AROW.

Funding: This material is based on work funded by the US National Science Foundation under ITR grants EIA 0205456 and IIS 0428193 and Career grant 0238295.

Conflict of Interest: none declared.

REFERENCES

Allen,J.E. and Salzberg,S.L. (2005) JIGSAW: integration of multiple sources of evidence for gene prediction. *Bioinformatics*, **21**, 3596–3603.

Allen,J.E. *et al*. (2004) Computational gene prediction using multiple sources of evidence. *Genome Res.*, **14**, 142–148.

Arumugam,M. *et al*. (2006) Pairagon+N-SCAN_EST: a model-based gene annotation pipeline. *Genome Biol.*, **7** (Suppl. 1), S5.1–S10.

Baumgartner,W.A., Jr *et al*. (2007) Manual curation is not sufficient for annotation of genomic databases. *Bioinformatics*, **23**, i41–i48.

Bernal,A. *et al*. (2007) Global discriminative learning for higher-accuracy computational gene prediction. *PLoS Comput. Biol.*, **3**, e54.

Brejová,B. *et al*. (2005) ExonHunter: a comprehensive approach to gene finding. *Bioinformatics*, **21** (Suppl. 1), i57–i65.

Burge,C.B. and Karlin,S. (1998) Finding the genes in genomic DNA. *Curr. Opin. Struct. Biol.*, **8**, 346–354.

Coghlan,A. *et al*. (2008) nGASP–the nematode genome annotation assessment project. *BMC Bioinformatics*, **9**, 549.

Crammer,K. *et al*. (2006) Online passive-aggressive algorithms. *J. Mach. Learn. R*, **7**, 551–585.

Crammer,K. *et al.* (2009) Adaptive regularization of weight vectors. In *Proc of NIPS*.

Dredze,M. *et al*. (2008) Confidence-weighted linear classification. In *ICML '08: Proceedings of the 25th International Conference on Machine Learning*, ACM, New York, NY, USA, pp. 264–271.

Elsik,C.G. *et al*. (2007) Creating a honey bee consensus gene set. *Genome Biol.*, **8**, R13.

ENCODE Project Consortium. (2004) The ENCODE (ENCyclopedia Of DNA Elements) project. *Science*, **306**, 636–640.

Guigo,R. and Reese,M.G. (2006) EGASP '05: ENCODE genome annotation assessment project. *Genome Biol.*, **7**.

Howe,K.L. *et al*. (2002) GAZE: a generic framework for the integration of gene-prediction data by dynamic programming. *Genome Res.*, **12**, 1418–1427.

Keibler,E. and Brent,M.R. (2003) Eval: a software package for analysis of genome annotations. *BMC Bioinformatics*, **4**, 50.

Liu,Q. *et al*. (2008) Evigan: a hidden variable model for integrating gene evidence for eukaryotic gene prediction. *Bioinformatics*, **24**, 597–605.

Pavlović,V. *et al*. (2002) A bayesian framework for combining gene predictions. *Bioinformatics*, **18**, 19–27.

Potter,S.C. *et al*. (2004) The Ensembl analysis pipeline. *Genome Res.*, **14**, 934–941.

Schiex,T. *et al*. (2001) EuGene: an eucaryotic gene finder that combines several sources of evidence. *LNCS 2066*, pp. 111–125.

Stanke,M. and Waack,S. (2003) Gene prediction with a hidden Markov model and a new intron submodel. *Bioinformatics*, **19** (Suppl. 2), II215–II225.

Stanke,M. *et al*. (2008) Using native and syntenically mapped cDNA alignments to improve de novo gene finding. *Bioinformatics*, **24**, 637–644.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016