Gene Prediction with Conditional Random Fields

Aron Culotta, David Kulp and Andrew McCallum
Department of Computer Science
University of Massachusetts
Amherst, MA 01002
{culotta, dkulp, mccallum}@cs.umass.edu

Abstract

Given a sequence of DNA nucleotide bases, the task of gene prediction is to find subsequences of bases that encode proteins. Reasonable performance on this task has been achieved using generatively trained sequence models, such as hidden Markov models. We propose instead the use of a discriminitively trained sequence model, the conditional random field (CRF). CRFs can naturally incorporate arbitrary, non-independent features of the input without making conditional independence assumptions among the features. This can be particularly important for gene finding, where including evidence from protein databases, EST data, or tiling arrays may improve accuracy. We evaluate our model on human genomic data, and show that CRFs perform better than HMM-based models at incorporating homology evidence from protein databases, achieving a 10% reduction in base-level errors.

1 Introduction

A common goal in bioinformatics is to infer the underlying systems that transform biological elements (e.g. DNA) into phenotypic expression (e.g. blue eyes). A central belief is that the structure of a biological element contributes to its function, and so a natural way to proceed is to collect elements with a common function and deduce the structural similarities responsible for this function.

For example, DNA is a sequence of nucleotide molecules (bases) which encode instructions for the generation of proteins. However, because not all of these bases contribute to protein manufacturing, it is difficult to determine which proteins will be generated from an arbitrary DNA sequence. Here, we can use *sequential structure* to determine which proteins, if any, a subsequence of bases will encode. We refer to these protein coding regions as *genes*; hence the task of *gene prediction* is to infer which subsequences of DNA correspond to genes.

A popular probabilistic sequence model is the hidden Markov model (HMM) [16], which has widespread use in natural language processing tasks such as speech recognition, part-of-speech tagging, and information extraction [4]. Noting the surprising similarities between DNA sequences and human language, bioinformatics researchers have successfully adapted HMMs to the task of gene prediction [6, 9].

HMMs are appealing models because it is relatively straightforward for a machine to learn their parameters and for a human to interpret them. However, one drawback of HMMs is that it is cumbersome to model arbitrary, dependent features of the input sequence.

This drawback of HMMs can be particularly troublesome for gene prediction, since there are many potentially valuable external sources of evidence (e.g. genomic databases, EST data, and tiling arrays). HMMs assume that each feature is generated independently by some hidden process; however, this is in general not the case. One way to address this problem is to explicitly model these dependencies by complicating the HMM structure, but this grows intractable as the number of features increases.

To overcome this, we propose the use of another sequence model that has become popular in language processing and vision tasks, the conditional random field (CRF) [10]. As opposed to the *generatively* trained HMM, the *discriminatively* trained CRF is designed to handle non-independent input features, which can be beneficial in complex domains. CRFs have out-performed HMMs on language processing tasks such as information extraction [15] and shallow parsing [21], and we show that similar performance gains can be obtained for gene prediction.

In addition to features computed over the given DNA sequence, we also incorporate homology features from the online genomic database BLAST [13]. We use the similarity between the given DNA sequence and sequences in protein databases as additional evidence for gene prediction.

We incorporate these features in both a CRF model and an HMM-based model, and find that CRFs achieve a 10% reduction in base-level errors.

2 Gene Prediction

2.1 Biological Background

In this section, we formalize the gene prediction task, beginning with a brief description of the biological phenomenon of protein synthesis.

Each chromosome in an organism contains a double-helix of nucleotide bases called deoxyribonucleic acid (DNA). Each nucleotide can be one of four molecules: adenine (a), cytosine (c), guanine (g), and thymine (t). Through the *transcription* process, the DNA helix is unwound and a contiguous subsequence of bases is copied to a new element called *pre-mRNA*. In eukaryotes, this *pre-mRNA* contains superflous sequences called *introns* that are not needed in subseqeunt processing. These introns are removed in the *splicing* stage, and the remaining bases (*exons*) are reassembled into *mRNA*.

The mRNA contains a contiguous subsequence of bases called the *coding sequence* (CDS) that are deterministically *translated* into amino acids. Translation maps base triples (*codons*) into amino acids, which combine to form a protein.

In gene prediction, we are given the original strand of DNA, and we must predict the CDS. Note that while protein synthesis is a multi-stage process, most gene prediction models (including the one presented here) model it as a one-stage process.

2.2 Notation

Let $\mathbf{x} = \{x_1 \ldots x_n\}$ be an input sequence of DNA bases, where $x_i \in \{a, c, t, g\}$. Each sequence $\mathbf{x}$ has an associated sequence of *labels* $\mathbf{y} = \{y_1 \ldots y_n\}$, where $y_i \in \{C, C'\}$ corresponds to *coding* or *non-coding* regions, respectively. The task of gene prediction is to find a mapping function such that $f(\mathbf{x}) = \mathbf{y}$.

3 Probabilistic Sequence Models

Probabilistic sequence models assume $\mathbf{x}$ and $\mathbf{y}$ are random variables and attempt to learn the statistical dependence between them.

3.1 HMMs

Hidden Markov Models (HMMs) are directed graphical models that define a factored probability distribution $p(\mathbf{x}, \mathbf{y})$, which, in a first-order model, decomposes as

$$p(\mathbf{x}, \mathbf{y}) = \prod_i p(x_i|y_i)p(y_i|y_{i-1}) \tag{1}$$

This is often referred to as a *generative* model, because the term $p(x_i|y_i)$ can be thought of as the probability that a label variable y_i "generates" the observation variable x_i. The second term, $p(y_i|y_{i-1})$, reflects the first-order Markov assumption that the probability of a label variable y_i is independent of all other labels given y_{i-1} (i.e. $I(y_i, \mathbf{y} \setminus y_{i-1}|y_{i-1})$).

Given a corpus $\mathcal{D}$ of labeled $(\mathbf{x}, \mathbf{y})$ pairs, maximum-likelihood training in an HMM consists of computing the values of $p(x_i|y_i)$ and $p(y_i|y_{i-1})$ that maximize the summed joint probability $\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} p(\mathbf{x}, \mathbf{y})$. In practice, this amounts to simply counting the frequency of each event and normalizing.

To infer the most likely $\mathbf{y}$ for an unlabeled sequence $\mathbf{x}$, the model chooses sequence $\mathbf{y}$ that maximizes $p(\mathbf{x}, \mathbf{y})$. This can be calculated efficiently with a dynamic program.

One limitiation of this model is that the observed variable x_i depends only on the label variable y_i. This means that when the model is predicting the value for label y_i, it cannot directly consider knowledge from observations $\mathbf{x} \setminus x_i$. To incorporate this knowledge in an HMM, we must expand the observation probability term, e.g. $p(x_1 \ldots x_n | y_i)$.

There are at least two problems with this modification. First, introducing a long-range dependence between, say, y_i and x_{i-5} implicitly adds dependence between label variables y_i and y_{i-5} (because now both y_i and y_{i-5} influence x_{i-5}). This precludes the nice factorization in Equation 1 and can make training and inference intractable. Second, estimating $p(x_1 \ldots x_n | y_i)$ from training data will likely suffer from data sparsity problems for all but the smallest values of n. This estimate requires a conditional probability table of size L^n, where L is the number of possible values each x_i can take.

Both of these problems are usually addressed by making conditional independence assumptions, e.g. $I(y_i, y_{i-5} | x_i, x_{i-5})$ for the first problem, and $I(x_i, \mathbf{x} \setminus x_i | y_i)$ (i.e. the “naive Bayes” assumption) for the second problem. However, these assumptions are often unrealistic and can lead to poor estimates of the observation probabilities.

3.2 CRFs

An alternative solution to this problem can be motivated from the following argument. HMMs model the *joint* distribution $p(\mathbf{x}, \mathbf{y})$. However, when predicting the labels of a new sequence $\mathbf{x}$, the only distribution needed is the *conditional* $p(\mathbf{y}|\mathbf{x})$. In other words, because $\mathbf{x}$ is always known at testing time, there is no need to model the uncertainty of $\mathbf{x}$ to perform prediction.

This solution can be captured by an *undirected graphical model* (also known as a *Markov random field*) [2]. In undirected graphical models, edges between nodes are no longer required to bear probabilistic semantics; rather, an edge simply represents some “compatibility function” between the values of random variables. These compatibility functions (or *potential functions*) are then globally normalized to assign a probability to each graph configuration. Because of this shift in semantics, we can add edges between label y_i and any set of observed nodes in $\mathbf{x}$ without creating additional dependencies among nodes in $\mathbf{y}$. Also, when $\mathbf{x}$ is observed, we can model $p(y_i|\mathbf{x})$ without considering the dependencies among $\mathbf{x}$.

A particular instance of an undirected graphical model in which a set of unobserved variables are conditioned on a set of observed variables is called a *conditional random field* (CRF) [10]. Figure 1 shows a CRF which makes a first-order Markov assumption among label variables $\mathbf{y}$, resulting in a linear-chain. Note that each label element of $\mathbf{y}$ has access to any of the observation variables in $\mathbf{x}$.

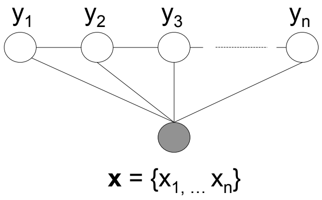

Figure 1: A first-order CRF with label variables $\mathbf{y}$ and observed variables $\mathbf{x}$

More formally, let $\mathcal{G}$ be an undirected model for $\mathbf{x}$ and $\mathbf{y}$, the fully connected subgraphs of which define a set of cliques $\mathcal{C} = \{\{\mathbf{x}_c, \mathbf{y}_c\}\}$. A CRF defines the conditional probability of label sequence $\mathbf{y}$ given observation sequence $\mathbf{x}$ as

$$p_\Lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in \mathcal{C}} \Phi(\mathbf{y}_c, \mathbf{x}_c; \Lambda) \tag{2}$$

where Φ is a real-valued potential function parameterized by Λ, and normalization factor $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{c \in \mathcal{C}} \Phi(\mathbf{y}_c, \mathbf{x}_c)$.

The potential function can be parameterized by an arbitrary set of feature functions $\{f_i\}$ over each clique, a common form of which is

$$\Phi(\mathbf{y}_c, \mathbf{x}_c; \Lambda) = \exp\Big(\sum_i \lambda_i f_i(\mathbf{y}_c, \mathbf{x}_c)\Big) \tag{3}$$

The model is parameterized by a set of weights $\Lambda = \{\lambda_i\}$, where each λ_i weights the output of feature function f_i. Note that in a first-order CRF, cliques contain labels y_i, y_{i-1} and an arbitrary subset of observations from $\mathbf{x}$. Thus, the prediction for label y_i is a function of the previous prediction y_{i-1} as well as any number of features over the *entire* input sequence $\mathbf{x}$.

Given a training corpus $\mathcal{D}$, maximum-likelihood training seeks to choose Λ such that the log-likelihood of the data is maximized, where the log-likelihood is given as

$$\mathcal{L}(\Lambda) = \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}} \log p_\Lambda(\mathbf{y}|\mathbf{x}) \tag{4}$$

This maximum can be found using gradient ascent methods, such as conjugate gradient or limited-memory BFGS [14].

It is important to note that CRF training maximizes the conditional likelihood (discriminitive training), whereas HMM training maximizes the joint likelihood (generative training). This is the source of the intuition that discriminitive training optimizes testing accuracy.

4 CRF Gene Model

We now describe the model topology and the features used to construct a CRF gene finder.

4.1 Finite-State Structure

A finite-state machine representation of a CRF provides restrictions on the possible transitions in the label sequence. These can be hand-crafted or inferred from a training corpus. Here, we use knowledge from the mechanics of protein synthesis to determine the finite-state structure.

We have previously described the set of possible labels as $y_i \in \{C, C'\}$, for coding and non-coding bases. To better reflect the underlying biological process, we expand this set to $y_i \in \{C, I, N\}$, where I is an intron, and N is either *intergenic* (a region not transcribed into pre-mRNA) or a non-coding exon.

Additionally, using the fact that proteins are encoded by base triples (codons), we must constrain our model to output coding regions whose length is a multiple of three. Thus, we expand label C to C_0, C_1, C_2, representing each *frame* in the codon, and also expand I to I_0, I_1, I_2 to store frame information across introns.

We also leverage regularities in the borders of coding regions (e.g. "splice sites"). Specifically, coding regions always begin with the start codon "atg" (in humans) and end in one of three stop codons: "taa", "tga", "tag." Furthermore, 99% of transitions from labels C to I begin with introns "gt", and transitions from I to C end with introns "ag" (the 5' and 3' splice sites, respectively). These are called *consensus dinucleotides*. These restrictions can be represented in the finite-state model by adding degenerate states that output a restricted set of bases (e.g. states InitialExon0, InitialExon1, InitialExon2 that only emit observations "a", "t", "g", respectively).

It is also known that the start codon cannot occur "in frame" anywhere within a coding region except at the beginning. To enforce this in a CRF, we construct a feature that is true if the current label is C_2, the previous label is C_1, and the previous three bases are the start codon "atg". To make this configuration impossible, we fix the weight for this feature to $-\infty$. (Note that this does not prevent the *true* start codon, since this is represented by the degenerate states described above.

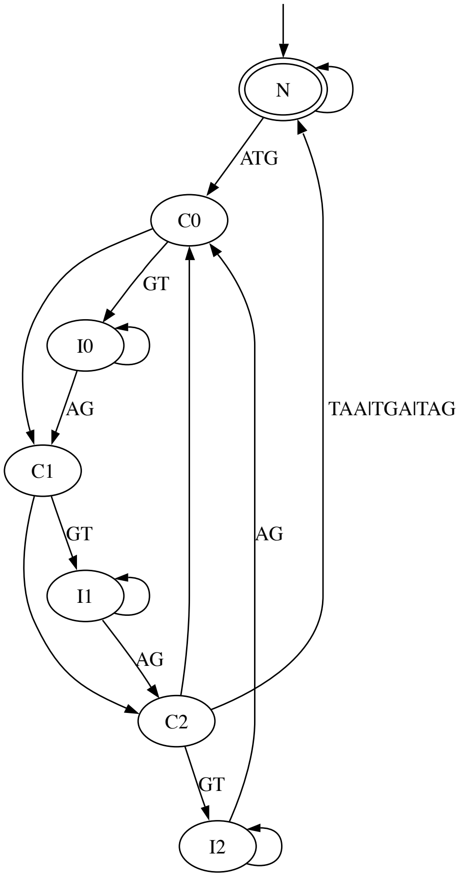

Figure 2: Finite-state machine for gene prediction. Edge labels represent base observations required for a transition. C represents coding states, I intronic states, and N intergenic and non-coding exon states.

Feature Type	SubTypes
Base	$b_{i-5}, b_{i-4}, b_{i-3}, b_{i-2}, b_{i-1}, b_i$ $b_{i-4}, b_{i-3}, b_{i-2}, b_{i-1}, b_i$ $b_{i+4}, b_{i+3}, b_{i+2}, b_{i+1}, b_i$ b_{i-2}, b_{i-1}, b_i b_{i+2}, b_{i+1}, b_i b_{i-1}, b_i b_{i+1}, b_i $b_{i-2}, b_{i-1}, b_i, b_{i+1}, b_{i+2}$ b_{i-1} b_{i+1} b_i

Histogram
windowSize=5, historySize=40,
futureSize=10
	frequency of base singletons, pairs, and triples
	frequency of disjunctions of size 2 (e.g. G or T)

Blastx	number of hits, maximum score, sum of scores conjunctions at $i-1 \wedge i$ and $i+1 \wedge i$ conjunctions with External features

Table 1: List of CRF features to predict label y_i, where b_i represents the identity of the base at position i.

Splice site boundary conditions are handled similarly.)

Figure 2 shows the finite state diagram for the CRF. For clarity, the degenerate states have been replaced with labeled edges indicating necessary conditions for transitions.

4.2 Feature functions

We construct a first-order CRF, using features of the input to capture the local and long-range dependencies among bases, as well as homology features obtained from a database of known proteins.

Since the coding regions consist of base triples that will be translated into amino acids, a useful feature is to examine the statistics of certain sequences of amino acids. This could be done with a hierarchical model that contains explicit states for types of amino acids, or by using a higher-order Markov model; however, similar information can also be captured by considering as a feature the identity of the previous five bases. Combined with the predicted label, this tells us the compatibility of pairs of amino acids. To capture local dependencies, we also include conjunctions of previous one, two, and three bases, as well as analagous features for the subsequent bases. We refer to this set of features as Conj.

It has also been observed [24] that splice sites have discriminitive signals dependent on approximately 40 prior bases and 20 subsequent bases. (e.g. the "branch site"). To capture these signals, we include *histogram features* which count frequencies of base conjunctions and disjunctions in sliding windows over bases upstream and downstream. For example, one feature is the number of times the base pair "g" or "c" occurred in a sliding window of size 5 in the previous 40 bases. We refer to these features as HISTOGRAM, the number of previous bases examined as HISTORYSIZE, the number of subsequent bases examined as FUTURESIZE, and the size of the sliding window as WINDOWSIZE.

Additionally, we include homology features by searching for similar DNA sequences in a database of proteins. Specifically, we issue BLAST queries over the "non-redundant" (NR) protein database to find similar DNA sequences that are already known to code proteins [13]. Since our test data may have exact matches in the BLAST database, we restrict hits to those with less than 70% identity to simulate performance on novel sequences.

These features capture weak homology of novel sequences to already sequenced genomes. Note that these features could be augmented to reflect a richer homology using the identity of the organisms with homologous sequences.

We calculate summary statistics for each base that quantify how often it appears in the BLASTX data and with what level of confidence.

A summary of all features is presented in Table 1.

5 Experiments

We test our model on a set of 450 human genes (over 5 million bases) from Genbank, release 105, 1998. The data is restricted to genes that (1) contain at least one intron, (2) begin with the start codon "atg", (3) end with one of the three stop codons, (4) have splice sites that conform to consensus dinucleotides, and (5) have no in-frame stop codons.We randomly split the data into 70% training and 30% testing.

Initially, CRF training proved difficult due to the large training set size and label imbalance problems (i.e. less then 10% of bases encode proteins). To alleviate this, we pruned the training dataset by limiting the length of contiguous sequences of I and N labels to 200. This was done by removing bases from the middle of these long sequences, preserving the input signals near splice sites.

We evaluate three CRF variants created by training with different feature sets. CRF + CONJ only uses the CONJ features; CRF + CONJ + BLASTX additionally uses BLAST homology features; and CRF + CONJ + HISTOGRAM + BLASTX additionally uses HISTOGRAM features.

We compare against two variants of the Genie system, [9], a fifth-order hid-

Model	Base Sp	Base Sn	F1
CRF + CONJ + HISTOGRAM + BLASTX	84.16	88.1	**86.09**
CRF + CONJ + BLASTX	79.61	86.63	82.97
CRF + CONJ	72.04	86.5	78.61
GENIE + BLASTX	85.43	83.68	*84.55*
GENIE	85.48	81.21	83.29

Table 2: Comparison of gene predictors. Note that CRFs better utilize the homology evidence given by BLASTX features

den *semi-Markov* model which has a finite-state structure similar to the CRF gene finder's.

The first variant (GENIE) trains the parameters of Genie using the same training DNA the CRF uses. The second variant (GENIE + BLASTX) constrains the output of Genie to agree with the coding regions predicted by a conservative use of Blastx data (i.e., regions with an "e-value" less than 10^{-30}). Note that both Genie variants incorporate the HISTOGRAM features into its prediction by generating profiles from labeled data.

We evaluate base-level performance of coding regions. *Specificity* (or *precision*) (Sp) is the percentage of predicted coding bases that are true coding bases. *Sensitivity* (or *recall*) (Sn) is the percentage of true coding bases that were predicted as such. $F1$ is the harmonic mean of these two measures ($\frac{2*Sn*Sp}{Sn+Sp}$). Results are displayed in Table 2.

The important comparison is between (CRF + CONJ + HISTOGRAM + BLASTX) and (GENIE + BLASTX). Here, we see that the CRF achieves a 10% reduction in F1 error over Genie, using the same data sources. This is indicative of the modeling power afforded by CRFs, which makes better use of the Blast data than does Genie.

GENIE and CRF + CONJ are similar models, since the 5th-order features of GENIE are captured by the CONJ features of the CRF. The difference in performance is likely due to the semi-Markov property of Genie, an issue discussed in Section 8, as well as the fact that GENIE incorporates the histogram features that CRF + CONJ does not. In Section 8, we discuss possible ways to improve CRF performance.

For all models, including BLASTX features improves performance, although more dramatically so for the CRF gene finder.

6 Related Work

There is a large amount of commercial and research software for gene prediction. For a more thorough treatment, see [17].

Generative methods based on hidden Markov model variants include Genie [9], GeneMark.HMM [11], and Glimmer [18].

More linguistically-motivated approaches include using context-free grammars (CFGs) to parse DNA sequences [20, 5]. While CFGs model the hierarchical properties of protein synthesis, our model flattens this representation from a CFG to a finite-state automaton.

Other methods are based on matching an unlabeled sequence against a database of known sequences. For example, a simple gene finder can be constructed by considering a number of possible protein coding sequences, translating the codons into protein sequences, then comparing them against a protein sequence database using, for example, BLASTX [13]. While this method will given reasonable approximations for highly conserved regions, it will perform poorly on more unusual sequences. In this paper, we have demonstrated a way to simply add these database to a CRF as external evidence.

A number of discriminatively trained methods have been used to predict local label configurations, which are then combined in a dynamic program to choose an optimal labeling for the entire sequence. For example, artificial neural networks have been used to predict splice sites [25, 22]. Also, linear [23] and quadratic [27] discriminant analysis has been performed on subsequences of bases. Similarly, maximum entropy classifiers have been used to classify splice sites [26], as well as in the related task of modeling amino acid sequences [3].

Note that these "local discriminative" methods share the CRF's capability of incorporating arbitrary features of the input sequence. Some of these methods, however, model adjacent label predictions independently; that is, they do not directly model $p(y_i|y_{i-1})$.

All of these local discriminative methods differ from CRFs in the method by which their parameters are learned. In a CRF, maximum likelihood training chooses weights to maximize the conditional probability of the entire sequence of labels given each training sequence. The global normalization term $Z(\mathbf{x})$ in Equation 2 allows interaction between weights from disparate locations in each sequence. Local discriminative methods, by contrast, decompose the conditional distribution into a product of local predictions, reducing this interaction. In general, this reduces training complexity as well as model capacity.

The works most relevant to ours are HMMgene and Hidden Neural Networks (HNNs) [7, 8]. HMMgene is an HMM model which is trained discriminatively. However, HMMgene still factors the joint distribution according to the semantics of a directed graphical model, and therefore cannot easily incorporate non-independent features. HNNs remedy this by constructing an undirected graphical model where the potential functions are multi-layer neural networks. If the potential functions instead were single-layer networks, HNNs would be equivalent to CRFs. To the best of our knowledge, results have not been reported using HNNs for gene prediction.

7 Conclusion

We have demonstrated that using external sources of information, such as BLAST protein databases, can improve results on gene prediction tasks. We have also shown that CRFs can incorporate this external evidence more effectively than traditional HMM models. The higher accuracy of CRFs is likely due to their weaker assumptions about the data and their discriminative training method.

These results suggest the promise both of incoporating disparate information sources for bioinformatics problems and of using undirected models like CRFs to leverage this evidence in a probabilistically coherent model.

8 Future Work

Two changes to the CRF model that will likely improve performance are (1) modeling the length distribution of coding regions and (2) incorporating richer features.

First, it has been shown that the length of exons tends to be similar within a species [9]. Furthermore, this length distribution is generally a right-skewed binomial. However, there is no guarantee that the CRF will produce such a distribution. To model this directly, we can use a semi-Markov CRF [19], where each state can now emit subsequences of observations, rather than single observations. By adding an extra dimension to the dynamic program, the semi-Markov CRF considers the length of predicted exons when choosing an optimal path.

Second, a major motivation for switching to discrimitively trained sequence models is to easily incorporate (1) long-distance features of the input and (2) externally generated evidence. Here, we have included long-distance features over the DNA sequence, but only limited external evidence. In addition to the BLAST features, other external features which may prove beneficial include examining databases of expressed sequence tags (ESTs) [1] or tiling array data. With CRFs, these external sources of evidence can be included without any knowledge of their dependency structure. Once these rich features are included, we could further employ automatic feature induction techniques to find non-obvious conjunctions of features that may improve performance [12].

9 Acknowledgements

This work was supported in part by the Center for Intelligent Information Retrieval, in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249, and in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0427594. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

References

[1] M.D. Adams, J.M. Kelley, J.D. Gocayne, M. Dubnick, M.H. Polymeropoulos, H. Xiao, C.R. Merril, A. Wu, B. Olde, and R.F. Moreno. Complementary DNA sequencing: Expressed sequence tags and human genome project. *Science*, 252(5013):1651–6, 1991.

[2] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, B36:192–236, 1974.

[3] Eugen C. Buehler and Lyle H. Ungar. Maximum entropy methods for biological sequence modeling. In *Workshop on Data Mining in Bioinformatics*, 2001.

[4] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, 1992.

[5] S. Dong and D. B. Searls. Gene structure prediction by linguistic methods. *Genomics*, 23(3):540–551, 1994.

[6] J. Henderson, S. Salzberg, and K.H. Fasman. Finding genes in DNA with a hidden Markov model. *Journal of Computational Biology*, 4(2):127–41, 1997.

[7] Anders Krogh. Two methods for improving performance of a HMM and their application for gene finding. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, pages 179–186, 1997.

[8] Anders Krogh and S.K. Riis. Hidden neural networks. *Neural Computation*, 11(2):541–563, 1999.

[9] David Kulp. *Protein-coding gene structure prediction using generalized hidden Markov models*. PhD thesis, University of California, Santa Cruz, 2003.

[10] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[11] A.V. Lukashin and M. Borodovsky. GeneMark.hmm: New solutions for gene finding. *Nucleic Acids Research*, 26:1107–1115, 1998.

[12] Andrew McCallum. Efficiently inducing features of conditional random fields. In *Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI03)*, 2003.

[13] S. McGinnis and T.L. Madden. BLAST: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Research*, 32:W20–W25, 2004.

[14] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 1999.

[15] Fuchun Peng and Andrew McCallum. Accurate information extraction from research papers using conditional random fields. In *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, 2004.

[16] L.R. Rabiner. A tutorial on hidden Markov models. In *IEEE*, volume 77, pages 257–286, 1989.

[17] M.G. Reese, G. Hartzell, N.I. Harris, U. Ohler, J.F. Abril, and S.E. Lewis. Genome annotation assessment in *drosophilia melanogaster*. *Genome Research*, 10:391–93, 2002.

[18] S. Salzberg, A. Delcher, S. Kasif, and O. White. Microbial gene identification using interpolated Markov models. *Nucleic acids research*, 26:544–548, 1998.

[19] Sunita Sarawagi and William W. Cohen. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing (NIPS17)*, 2004.

[20] D. B. Searls. The linguistics of DNA. *American Scientist*, 80:579–591, 1992.

[21] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In Marti Hearst and Mari Ostendorf, editors, *HLT-NAACL: Main Proceedings*, pages 213–220, Edmonton, Alberta, Canada, 2003.

[22] E.E. Snyder and G.D. Stormo. Identification of protein coding regions in genomic DNA. *Journal of Molecular Biology*, 248:1–18, 1995.

[23] V.V. Solovyev. Identification of human gene structure using linear discriminant functions and dynamic programming. *ISMB*, 3:367–375, 1995.

[24] G. D. Stormo. Consensus patterns in DNA. *Methods Enzymol*, 183:211–21, 1990.

[25] E.C. Uberbacher, U. Xu, and R. J. Mural. Discovering and understanding genes in human DNA sequence using GRAIL. *Methods Enzymol*, 266:259–281, 1996.

[26] Gene Yeo and Christopher B. Burge. Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals. In *Proceedings of the seventh annual international conference on Computational molecular biology*, pages 322–331. ACM Press, 2003.

[27] M.Q. Zhang. Identification of protein coding regions in the human genome based on quadratic discriminant analysis. In *Proceedings of the National Academy of Science*, volume 94, pages 565–68, 1997.