A Meta-state HMM with application to gene structure identification in eukaryotes

Stephen Winters-Hilt[1]* & Carl Baribault[1]

[1] University of New Orleans, New Orleans, LA 70148

*Corresponding author: winters@cs.uno.edu

Abstract

We introduce a generalized-clique hidden Markov model (HMM) and apply it to gene finding in eukaryotes (*C. elegans*). Our objective with the clique generalization is to improve the modeling of the critical signal information at the transitions between exon regions and non-coding regions, e.g., intron and junk regions. In doing this we will arrive at a HMM structure identification platform that is novel and robustly-performing in a number of ways.

The generalized clique HMM begins by enlarging the primitive hidden states associated with the individual base labels (as exon, intron, or junk) to substrings of primitive hidden states or *footprint* states. The emissions are likewise expanded to higher order in the fundamental joint probability that is the basis of the generalized-clique, or 'meta-State', HMM. We then consider application to eukaryotic gene finding and show how a meta-state HMM improves the strength of coding/noncoding-transition contributions to gene-structure identification. We will describe situations where the coding/noncoding -transition modeling can effectively 'recapture' the exon and intron heavy tail distribution modeling capability as well as manage the exon-start 'needle-in-the-haystack' problem. In analysis of the *C. elegans* genome we show that the sensitivity and specificity (SN,SP) results for both the individual-state and full-exon predictions are greatly enhanced over the standard HMM when using the generalized-clique HMM.

Introduction

Computational gene-finding dates back to the 1980's [1-3]. The most successful gene-finding tool has been the hidden Markov model, both in statistics intrinsic to the genome under study (*ab initio* gene-finding) [1-3], and in statistical analysis extrinsic to the genome (homology or EST matching) [4]. Matching, or alignment, of query sequences to a known sequence database is typically done using BLAST [5] (which involves an HMM seed-alignment, followed by less optimal, but faster, non-HMM seed-alignment extension). BLAST can also be used for gene finding alone, in homology-based programs to identify new genes by sufficiently aligning a query sequence with a known gene or genes [4]. In [6], they combine homology information with intrinsic genomic information (from statistical properties of the genomic sequence data alone). The main drawback of homology-based approaches is that they appear to be very weak at finding new genes, as discussed in [1], and explored in [7]. This is largely because approximately half of the genes in eukaryotic genomes appear to be novel to that genome (such as for *C. elegans*). This is likely to be true for humans, where we already know that only 50% of the proteins encoded in chromosome 22, for example, are found to be similar to previously known proteins. In [8], the author describes application of the best gene-finders known at the time (c.a. 2004) to gene-finding in novel genomes. From that study it is clear that gene-prediction is species-specific, i.e., an *ab initio* component must operate for any gene-finder to succeed at identifying genes and genomic structures novel to that organism [8].

Beginning c.a. 2000 there was a movement towards consolidation of the intrinsic and extrinsic approaches [7,9], as described in a 2002 review [9] and a 2006 review [10]. Furthermore, in the 2006 review, it was claimed that "improved modeling efforts at the hidden Markov model level are of relatively little value." We describe here a radical improvement in HMM capabilities in gene-finding, and likely a number of other areas of application, by introducing a fundamental new development at the model level. Also beginning c.a. 2000 was specialization to sensor development [11-17] to help supplement the HMM-based structure discovery process. There were sensors for transcription start site prediction [6], transcription initiation sites and polyadenylation signals [18], splice-site recognition [19,20], and identification of 3' ends of exons by EST analysis [21], to list just a few examples.

The past decade, since 2000, has also seen rapid growth in motif-discovery algorithms -- in parallel with the aforementioned sensor specialization (and growing more interdependent, as we describe in the Discussion). Many of these motif-discovery algorithms are beginning to tie into the HMM-based structure identification via referencing regions indicated by the HMM. In [22] and [23], many important TFBS's, promoters, and other regulatory motifs can be identified by their position relative to the start and stop of coding (and other non-self transitions identified by the HMM's optimal Viterbi-path parsing). In [22] they find that the motif finding effort is greatly enhanced by referencing to nearby gene-structure and identifying "peak regions" where motifs can be isolated. Not surprisingly, if separate statistical profiling is performed on the regions just outside (before and after) the transcription region, then gene-finding is improved [22,24]. Motif discovery can be focused onto the cis-regulatory regions in particular, and if linked with the HMM discovery, the motif-discovery and gene-discovery efforts are simultaneously strengthened. One of the clear benefits of having a very strong intrinsic HMM formulation as a foundation is that the later pairing with motif discovery and signal-sensor augmentations then

arrives at a unified and powerful intrinsic/extrinsic gene and motif discovery platform. This is capability is enhanced further if zone-dependent emissions are employed via larger meta-states (see Discussion) or via reference to HMMD improvements as indicated in [24-26]. The HMM formulation with HMMwD augmentation also provides an optimal means for inclusion of extrinsic statistics (side-information) into the Viterbi optimization (as described in [24]). The 'scaffolding' provided by the HMM parsing (via the Viterbi path derivation) defines regions where zone-dependent statistics and zone-restricted motif-discovery can be applied. Many motif-finding methods would benefit from the alignment referencing provided by the HMM's scaffolding of annotation across coding and non-coding regions. With zone-restricted motif discovery, gap and hash interpolated Markov model's [27,28] become powerful tools for motif discovery in a restricted region [18,28-32]. The approach we describe in this paper, and its companion paper [24], seeks to unify the above approaches within a powerful new HMM-based structure-modeling architecture.

The shortcomings of the HMM due to algorithmic definitions, such as lack of state-duration modeling, are readily apparent (with fixes as described in [24-26]). The shortcomings of the HMM due to model definition and related implementation, are more subtle. In an HMM implementation the number of look-ups to a particular emission or transition probability 'table' will show how that table's anomalous statistics influence the overall computation (where the count on use of a particular *component* in the table is precisely what provides an estimation in the HMM Baum-Welch algorithm). Similarly, what is readily observed in implementation of an HMM is the use of various probability tables, and a significant shortcoming is revealed. Standard HMM's lead to a model that strongly de-emphasizes (low table usage) and does not recognize the anomalous statistics known to exist around non-self transitions, and fundamentally, their transition probabilities are not sequence dependent. In this paper we demonstrate use of transition probabilities that are sequence dependent, via use of a constrained set of 'meta-states', with comparable computational complexity to the standard HMM. There is, thus, a 'choice in model primitives' shortcoming underlying the standard HMM implementations that is resolved in the meta-state HMM description to follow.

In this paper we introduce a generalized-clique, 'meta-state', hidden Markov model, and apply it to the analysis of the genomic structure of *C. elegans* (a genome-data intrinsic approach, e.g., not using EST or homology information). Our meta-state HMM generalizes from primitive states to windows of adjacent primitive states (e.g., "footprint states"), and does so by only allowing one coding-to-noncoding, or noncoding-to-coding, transition in the window of states. The constraint to have no more than a single 'non-self' transition in a footprint is equivalent to a minimum length constraint on exons, introns, and 'junk'. The linear growth in higher order states with this constraint (proven later) is critical for practical use of the larger footprint size models that will be demonstrated.

The generalized clique HMM begins by enlarging the primitive hidden states associated with individual base labeling (as exon, intron, or junk) to substrings of primitive hidden states or *footprint* states -- 'ieeeeeee' for example (also a Cajun exclamation). In what follows, the transitions between primitive hidden states for coding {e} and non-coding {i,j}, {ei,ie,je,ej}, are referred to as 'eij-transitions', and the self transitions, {ee,ii,jj}, are referred to as 'xx-transitions'. The emissions are likewise expanded to higher order in the fundamental joint

probability that is the basis of the generalized-clique, or 'meta-State', HMM. We consider application to eukaryotic gene finding and show how a meta-state HMM improves the strength of eij-transition contributions to gene-structure identification. We will describe situations where the meta-state eij-transition modeling can effectively 'recapture' the exon and intron heavy tail distribution modeling capability as well as manage the exon-start 'needle-in-the-haystack' problem.

Background

Genomic Data – with *C. elegans* specifics

Once it is fully annotated, genomic data can be unambiguously represented by strings formed from the 4 letters a, c, g, and t denoting the DNA nucleotide bases adenine, cytosine, guanine, and thymine, respectively. Genes are sequences of DNA nucleotides that encode specific sequences of amino acids to form proteins (with 5' to 3' read convention). The data annotation designates the coding and non-coding segments in the genomic data. In eukaryotes, genes consist of coding segments or exons which are delimited internally by special, intragenic, non-coding segments or introns. The *inter*genic, non-coding regions of bases outside the genes are referred to here as 'junk'.

The process of removing the intermediate introns and reconnecting (possibly variable subsets of) the resulting exons end-to-end is referred to as *splicing*. Perhaps the most important role of introns is to provide a mechanism for the formation of alternative combinations and/or subsets of the exons contained in a given gene in order to form alternative proteins also used by the organism in question. These alternative combinations are referred to as *alternative splicings*.

The *C. elegans* genome consists of six chromosomes {I,II,III,IV,V,X}, containing approx. 97,000,000 base-pairs of DNA. The 90% base accuracy of our meta-state HMM is sufficient to isolate and resolve outrons and other structures [33], such as the following dozen attributes:

(1) Approx. 19,000 genes, so approx. 1 gene per 5,000 bases.
(2) Each gene has an average of 5 introns.
(3) Tandem repeats account for 2.7% of genome, inverted repeats 3.6%. Repeats have different families on different chromosomes, and are more likely on introns. Common TTAGGC hexamer repeat.
(4) 38 dispersed repeat families can potentially be identified via hash interpolated Markov model [27].
(5) Approx. 50% of genome novel.
(6) Approx. 80% of genes are trans-spliced to a common spliced leader.
(7) Approx. 20% of genes organized as operons.
(8) Common occurrence of 'outron' structure: introns-like sequence with no internal 5' consensus that is found before the first exon.
(9) Genes with trans-splices are often distinguished from those that are not by the prescence of an outron.
(10) 3' ends of genes within operons typically singaled by AATAA.
(11) Typical translation Initiation: [(A/G)CCATG]
(12) Termination (TAA (61%); TAG (17%); TGA (22%))

The standard 1st Order HMM

We define the 1st order HMM as consisting of the following:

- An observable alphabet, B
- A hidden state alphabet, Λ
- "Prior" Probabilities $P(\lambda)$ for all $\lambda \in \Lambda$
- "Transition" Probabilities $P(\lambda_2|\lambda_1)$ for all $\lambda_1\ \lambda_2 \in \Lambda$ -- where the standard transition probability is denoted $a_{kl} = P(\lambda_n=l|\lambda_{n-1}=k)$, a 1st order Markov model on states with homogenous stationary statistics (i.e., no dependence on position 'n').
- "Emission" Probabilities $P(b|\lambda)$ for all $\lambda \in \Lambda$ $b \in B$ – where the standard emission probability is $e_k(b) = P(b_n=b|\lambda_n=k)$, a 0th order Markov model on bases and with homogenous stationary statistics.

Given the above, there are three classes of problems which the HMM can be used to solve [34,35]:

1. Evaluation - Determine the probability of occurrence of the observed sequence.
2. Learning - Determine the most likely emissions and transition.
3. Decoding (Viterbi) - Determine the most probable sequence of states emitting the observed sequence.

Here we focus only on the 3rd problem, the Viterbi decoding problem. The probability of a sequence of observables $B=b_0\ b_1 \ldots\ b_{n-1}$ being emitted by the sequence of hidden states $\Lambda=\lambda_0\ \lambda_1 \ldots\ \lambda_{n-1}$ is solved by using $P(B, \Lambda) = P(B|\Lambda)\ P(\Lambda)$ in the standard factorization, where the two terms in the factorization are described as the *observation model* and the *state model*, respectively. In the 1st order HMM, the state model has the 1st order Markov property and the observation model is such that the current observation, b_n, depends only on the current state, λ_n:

$$P(B|\Lambda)\ P(\Lambda) = P(b_0|\lambda_0)\ P(b_1|\lambda_1)\ldots P(b_{n-1}|\lambda_{n-1}) \times P(\lambda_0)P(\lambda_1|\lambda_0)P(\lambda_2|\lambda_0, \lambda_1)\ldots P(\lambda_{n-1}|\lambda_0 \ldots \lambda_{n-2})$$

With first order Markov assumption in the state-model this becomes:

$$P(B|\Lambda)\ P(\Lambda) = P(b_0|\lambda_0)\ P(b_1|\lambda_1)\ldots P(b_{n-1}|\lambda_{n-1}) \times P(\lambda_0)P(\lambda_1|\lambda_0)P(\lambda_2|\lambda_1)\ldots P(\lambda_{n-1}|\lambda_{n-2})$$

In the Viterbi algorithm, a recursive variable is defined (following the notation in [34]): $v_k(n) =$ *"the most probable path ending in state 'k' with observation 'b_n' "*. The recursive definition of $v_k(n)$ is then: $v_l(n+1) = e_l(b_{n+1})\ \max_k\ [v_k(n)\ a_{kl}]$. From which the optimal path information is recovered according to the (recursive) trace-back:

$$\Lambda^* = \text{argmax}_{\ \Lambda}\ P(B, \Lambda) = (\lambda^*_0, \ldots, \lambda^*_{n-1})$$

$$\lambda^*_n\big|_{\lambda^*_{n+1}=l} = \text{argmax}_k\ [v_k(n)\ a_{kl}], \text{ and where } \lambda^*_{L-1} = \text{argmax}_k\ [v_k(L-1)], \text{ for length L sequence.}$$

HMM states for gene-structure identification

Exons have a 3-base encoding as directly revealed in a mutual information analysis of gapped base statistical linkages as shown in [27]. The 3-base encoding elements are called *codons*, and the partitioning of the exons into 3-base subsequences is known as the codon *framing*. A gene's

coding length must be a multiple of 3 bases. The term *frame position* is used to denote one of the 3 possible positions – 0, 1, or 2 by our convention – relative to the start of a codon. Introns may interrupt genes after any frame position. In other words, introns can split the codon framing either at a codon boundary or one of the internal codon positions.

Although there is no notion of framing among introns, for convenience we associate framing with the intron, as indicated in the example below, as a tracking device in order to ensure that the frame of the following introns-to-exon transition is constrained appropriately. The primitive states of the individual bases occurring in exons, introns, and junk are denoted by:

Exon states = { e_0, e_1, e_2 },
Intron states = { i_0, i_1, i_2 },
Junk state = {j}.

We have three possible intron framings indicated in the following state strings.

jj...j$e_0e_1e_2$…e_0**i_0i_0…i_0**e_1…$e_0e_1e_2$jj…j (intron frame 0)
jj...j$e_0e_1e_2$…e_1**i_1i_1…i_1**e_2…$e_0e_1e_2$jj…j (intron frame 1)
jj...j$e_0e_1e_2$…e_2**i_2i_2…i_2**e_0…$e_0e_1e_2$jj…j (intron frame 2)

There are 15 unique two-label (dimer) transitions: {jj, je_0, e_0e_1, e_1e_2, e_2e_0, e_0**i_0**, e_1**i_1**, e_2**i_2**, **i_0i_0**, **i_1i_1**, **i_2i_2**, **i_0**e_1, **i_1**e_2, **i_2**e_0, e_2j}. In what follows we split the stop codon into the three possibilities strictly observed {e_2j_TAA, e_2j_TAG, e_2j_TGA }, for a total of 17 states in our forward encoding model.

Encodings for proteins can be found in both directions along the DNA strand. The encodings are sparse, rarely overlapping, and have approximately equal numbers of forward and reverse ('shadow') encodings. The differences in the base statistics in the forward and reverse gene encodings are sufficiently negligible (or disjoint) that their counts can simply be merged in the modeling (data not shown). We incorporate *shadow* states, indicating reverse encoded exons and introns, into the state model of our meta-state HMM, denoted by the primitives by ê and î, respectively. For example, the 3 possible intron framings for the reverse encoding are as follows:

jj...j$ê_2ê_1ê_0$…$ê_1$**$î_0î_0$…$î_0$**$ê_0$…$ê_2ê_1ê_0$jj…j (intron frame 0)
jj...j$ê_2ê_1ê_0$…$ê_2$**$î_1î_1$…$î_1$**$ê_1$…$ê_2ê_1ê_0$jj…j (intron frame 1)
jj...j$ê_2ê_1ê_0$…$ê_0$**$î_2î_2$…$î_2$**$ê_2$…$ê_2ê_1ê_0$jj…j (intron frame 2)

There are 16 reverse encoding state transitions in direct correspondence with the 16 non-jj state transitions for the forward read. The jj transition couples the forward and reverse reads in that a forward encoding can 'end', i.e., transition to a region of junk, then eventually transition to a reverse encoded gene. The total number of state-transition (dimer states) in our model is, thus, 33:

13 xx-type (homogeneous) dimers
a. 6 Intron-intron – i_0i_0, i_1i_1, i_2i_2, $î_0î_0$, $î_1î_1$, $î_2î_2$
b. 6 Exon-exon – e_0e_1, e_1e_2, e_2e_0, $ê_0ê_1$, $ê_1ê_2$, $ê_2ê_0$

c. 1 Junk-junk – jj

20 eij-type (heterogeneous) dimers

d. 6 Exon-intron – $\mathtt{e}_0\mathtt{i}_0$, $\mathtt{e}_1\mathtt{i}_1$, $\mathtt{e}_2\mathtt{i}_2$, $\hat{\mathtt{e}}_0\hat{\mathtt{i}}_0$, $\hat{\mathtt{e}}_1\hat{\mathtt{i}}_1$, $\hat{\mathtt{e}}_2\hat{\mathtt{i}}_2$
e. 6 Intron-exon – $\mathtt{i}_0\mathtt{e}_1$, $\mathtt{i}_1\mathtt{e}_2$, $\mathtt{i}_2\mathtt{e}_0$, $\hat{\mathtt{i}}_0\hat{\mathtt{e}}_1$, $\hat{\mathtt{i}}_1\hat{\mathtt{e}}_2$, $\hat{\mathtt{i}}_2\hat{\mathtt{e}}_0$
f. 6 Exon-junk – $(\mathtt{e}_2\mathtt{j})_{TAA}$, $(\mathtt{e}_2\mathtt{j})_{TAG}$, $(\mathtt{e}_2\mathtt{j})_{TGA}$, $(\hat{\mathtt{e}}_2\mathtt{j})_{TAA}$, $(\hat{\mathtt{e}}_2\mathtt{j})_{TAG}$, $(\hat{\mathtt{e}}_2\mathtt{j})_{TGA}$
g. 2 Junk-exon – $(\mathtt{j}\mathtt{e}_0)$, $(\mathtt{j}\hat{\mathtt{e}}_0)$

In order to work directly with the above dimer states, or the footprint-state generalization introduced in the Methods, we need to generalize to a higher order HMM model. The standard HMM has emissions that only dependent on the current state (e.g., we have $P(b_{n-1}|\lambda_{n-1})$ terms). This leads to poor performance in modeling the anomalous statistics in the transition regions between exon, intron, or junk regions. If a transition '$\mathtt{j}\mathtt{e}_0$' has occurred, for example, and we are looking at the base emission for the '$\mathtt{e}_0$' state, we can't account for the prior state with the simple $P(b_{n-1}|\lambda_{n-1})$ conditional probabilities in the standard bare-bones HMM modeling, we minimally need $P(b_{n-1}|\lambda_{n-2}, \lambda_{n-1})$, i.e., state modeling at the dimer-level or higher.

Methods

The Methods section begins with a description of the Dataset preparation in Sec. (1) titled 'Selection and Preparation of Data Sets….'. Sec. (2), on 'Application of meta-state HMM model to the Test Data' provides an overview of how the datasets and meta-state HMM models are used in the testing and tuning. In Sec. (3) on 'The Generalized-clique HMM Construction', we provide the core new HMM theory that is the underpinning of the new type of HMM modeling enabled. Section (4) gets into the nuts-and-bolts of the 'Enumeration of the Footprint States' in the meta-state HMM, and Sec (5) to follow provides the 'Measures of Predictive Performance that are used'.

(1) Selection and Preparation of Data Sets for preliminary testing and 'raw' Genome analysis

In [16], the authors performed the following steps to arrive at the ALLSEQ data set [37]:

1) Select the set of all sequences encoding at lease one complete protein from the vertebrate divisions of GenBank Release 85.0 (October 15, 1994).
2) From the above discard the following:
    a. Any sequence encoding at least one incomplete protein.
    b. Any sequence for which the exact coding regions was not unambiguous.
    c. Any sequence encoding a protein in the complementary (reverse encoding) strand.
    d. Any sequence containing a gene or part of a gene associated with other sequences.
    e. Any sequence encoding a pseudogene (via "CDS Key'' value ''/pseudo'').
    f. Any sequence encoding more than one gene or alternative splicing of a gene.
    g. Any sequence encoding a gene without introns.
3) From the 1410 sequences resulting from the above the following further discards were made:
    a. Any sequence whose coding segment did not start with the start codon ATG.

b. Any sequence whose coding segment did not end with a stop codon (TAA, TAG, TGA).
c. Any sequence whose coding segment was not a multiple of 3 in length.
d. Any sequence with any intron not beginning with GT and/or ending with AT (sic).
e. Any sequence whose coding segment contained an in-frame stop codon.

4) The following additional discards were made:
   a. Sequences for immunoglobulins, histocompatibility antigens and additional pseudogenes not discarded using previous criteria.
   b. 3 sequences longer than 50,000 bp.
5) One final selection was made from the sequences surviving the above in that the sequence's date of entry postdated Release 74 of Genbank (January, 1993) – intended as such to minimize the overlap of the resulting test set with training sets for the programs tested in [16].

As mentioned previously, because the training and testing sets were identical in our case, or close to identical in the Burset and Guigo study [14,16], we consider the ALLSEQ results as a brute force parameter search yielding what to expect in the ideal case and not necessarily a valid test of prediction performance. (The authors in [16] separate the test set from the training set by a date of entry criterion, but there was significant overlap between the testing and training data sets obtained [14] (an inevitable overlap since the ALLSEQ data set consisted of the "vast majority" of vertebrate sequences available at the time). We compare our initial test results with those reported by Burset and Guigo for this reason.

Early gene finding efforts are described in [38-40]. The authors of [14] provide an informative discussion, and references, on exon and intron durations, among other things. In [38], the authors observe "that the in-phase hexamer measure, which measures the frequency of occurrence of oligonucleotides of length six in a specific reading frame, is the most effective" for inclusion in gene finding. Moreover, those authors assembled their own test data set, called HMR195 [41], based on sequences submitted to Genbank after August 1997. We proceed with the results of the clique-parameter search using the ALLSEQ dataset. The ALLSEQ dataset properties are summarized in Table 3.

# Bases	Coding Density	Sequences			Introns			Exons		
		Total	BP	Avg. Len.	Total	BP	Avg. Len.	Total	BP	Avg. Len.
2892149	0.15	570	1754950	3078.86	2079	1310452	630.33	2649	444498	167.80

Table 3. Properties of the ALLSEQ data set

Five-fold cross-validation on single encoding (non-alternatively spliced) regions of Chromosomes I-V of C. elegans:

The data for Chromosomes I-V of *C. elegans* were obtained from release WS200 of Wormbase [42]. We note that the sixth and final chromosome of *C. elegans*, designated for legacy reasons as Chromosome X, was excluded from this analysis as it is known to have substantial differences in gene encoding properties as compared to Chromosomes I-V.

The following steps were used in order to prepare the data (described in Tables 4 & 5) prior to training and testing.

1) The data was scanned for in-frame stops, and ultimately no in-frame stops were detected.
2) The data was scanned for alternative splicing, and 6260 (30.5%) out of a total of 20514 sequences represent alternative splicing – including some forward encoded alternative splicings overlapping with reverse encoded alternative splicings.
3) In order to avoid the complexities involved in the prediction of alternative splicings, the *transitive closure* with respect to overlap of all alternative splicings was deleted from the data and the remaining annotation was appropriately offset in compensation for the deletions. For all branches of all alternative splicing sequences – along with any sequences interfering with them - the following segments, s, were deleted:
    a. s=the 5'- UTR, where (15b< length(s) <=200b) (15=WS/2: See item 7 below)
    b. s=the 3'- UTR, where (15b< length(s) <=3kb), and
    c. s=the entire coding sequence, CDS, including exons and introns
4) In order to avoid both the complexity of segmented prediction as well as any bias toward any specific subset of chromosomes during cross-validation, the following were performed:
    a. Both data and annotation files for all 5 chromosomes were divided into a total of 67 autonomous chunks of nominal size 1Mb and minimum size 500kb.
    b. The resulting 67 chunks were then evenly (as allowable) distributed into five (5) groups for 5-fold cross-validation.
5) Training was performed independently on each of the above chunk groups with a sampling window size of first WS=30, then WS=40.
6) Five-fold cross-validation counts from training on chunk groups 1-4 were combined to form probability estimates used to test on chunk group 5, then training on 2-5 for testing on 1, and so on.

Summary of data reduction in *C. elegans*, Chromosomes I-V						
File	# sequences	# alt.	% alt.	# exons	# alt.	% alt.
CHROMOSOME_I	3537	1306	36.92%	24295	10942	45.04%
CHROMOSOME_II	4161	1316	31.63%	25427	10427	41.01%
CHROMOSOME_III	3277	1220	37.23%	21541	9614	44.63%
CHROMOSOME_IV	3886	1195	30.75%	24390	9509	38.99%
CHROMOSOME_V	5653	1222	21.62%	32135	9122	28.39%
Total	20514	6259	30.51%	127788	49614	38.83%

Table 4. Summary of data reduction in *C. elegans*, Chromosomes I-V.

# Bases	Coding Density	Sequences			Introns			Exons		
		Total	BP	Avg. Len.	Total	BP	Avg. Len.	Total	BP	Avg. Len.
67000811	0.24	14255	32547117	2283.2	63919	16371001	256.1	78174	16176057	206.9

Table 5. Properties of data set *C. elegans*, Chromosomes I-V (reduced)

Note: sequence-BP – (intron-BP + exon-BP) = 59, due to a premature start of the sequence ZK1010.9 of Chromosome III in the annotation provided.

(2) Application of meta-state HMM model to the Test Data

The meta-state HMM is higher order in both base-emission Markov order and state-transition Markov order, i.e., the meta-state HMM describes an irreducible joint-probability, or 'clique', generalization. The footprint states created from windows of 13 primitive states (or footprint size F=12, in consecutive overlapping 'dimers') lead to one of our best performing models, with *full-exon* predictive accuracy of 86% on the B&G ALLSEQ data [16] (with data used as both train and test for comparison with GeneID+ and FGENEH). One method, FGENEH, is similar to ours in that it only uses the intrinsic genomic sequence data (not homology searches, etc.). FGENEH's predictive accuracy on the same ALLSEQ data was 64% [16]. One of the best scoring methods on the ALLSEQ data is GeneID+, whose accuracy is 71%, where GeneID+ *does* use external information [16]. The base-level accuracy of our meta-state HMM on the ALLSEQ data is 97%, compared to 86% scoring at the full-exon correct level, indicating that improvement in identification of coding/non-coding transitions would improve results, particularly at start-of-coding. This has been addressed in [17] with the introduction of SVM methods so won't be elaborated upon here. Further efforts to merge the SVM sensor into the meta-state HMM are described in the Discussion.

Other gene finding methods typically involve some degree of pre-processing – as is made clear by how their test-data is often arranged (e.g., the 570 *separate* sequences, each containing one gene, in the B&G ALLSEQ dataset [16]). When examining these datasets, and then turning to applying our methods on large blocks of genomic data, there seems to be a 'contrast' problem in the recognition of the start-of-coding region when working with the standard 1^{st}-order HMM (a 'needle-in-the-haystack' problem). We find in our meta-state HMM approach that the contrast problems are automatically solved, and that many of the beneficial attributes of HMM-with-duration modeling are, remarkably, recovered (the heavy-tail modeling capability on intron and exon length distributions in particular).

In this effort we also wanted to introduce a new dataset that minimally alters the full genome dataset. We want our optimized HMM to also lay the foundation for a multifaceted regulatory motif discovery process. The gene prediction, in the end, will not only identify gene-structure, but it will have done so by identifying similar structures and regions in relation to the eij-transitions. The regions around the predicted eij-transitions can, thus, be analyzed using focused motif-finder approaches (like the MI method in [27] and [28], to then decipher various aspects of gene-regulation). To this end, our main concern with the raw *C. elegans* genomic data is that the alternatively spliced regions will be harder for the HMM to manage, since it is not part of the modeling in any way, and will be harder to score, since one prediction will exclude an overlapping alternatively-spliced variant, such that to be correct on one you have to be wrong on the other. So our approach is to simply drop the regions of the genome that have alternatively splice genes. More precisely, we drop those segments of the genome corresponding to the transitive closure with respect to overlap of alternatively spliced genes. The alternatively-spliced regions are simply dropped from the working dataset (resulting in dataset *C.elegans* reduced), and the annotation is offset as needed to compensate for the deletions. The alt-splice redacted set of genomic data that we obtain is reduced by 30.5% for Chromosomes I-V (*C. elegans* genome release WS200). We make no use of the sixth chromosome (labeled as X, roman numeral ten, for legacy reasons), where the odd naming convention is the least of the oddities of this chromosome, which has a large contribution from non-protein encoding DNA (tRNA, etc.).

Our alternative-splice redacted *C. elegans* genome has chromosomes I-V concatenated, then split into 67 non-overlapping chunks, which are then evenly distributed (as allowable) amongst five groups ('folds'). Five-fold cross-validation was then performed: where 4-folds are used in learning the HMM parameters, and the other fold used to test, with prediction scored against the annotation on that fold, and this process repeated with other folds held out, then averaged over all five cross-validations to obtain the prediction accuracies detailed in the Results. On the alt-splice redacted genome we have a full-exon prediction accuracy of 74% (with F=20), while the F=2 model, with minimal footprint, has full-exon predictive accuracy of only 61%, in rough agreement with the performance of standard-HMM gene finders with purely intrinsic information (like FGENEH). The base level accuracy at F=20 is 90%, so as with the ALLSEQ data, there is clear room for improvement with better eij-transition recognition. Further details are left to the Results Section, along with Discussion and Conclusion. In the Methods Section we describe: (i) dataset preparation; (ii) generalized HMMs; (iii) the generalized footprint state structure for gene-finding; and (iv) the measures of accuracy used. In the Background that follows we describe (i) the data to be analyzed; (ii) HMMs; and (iii) HMMs with state structure for gene-finding.

(3) The Generalized-clique HMM Construction

The traditional HMM assumes that a 1st order Markov property holds among the states and that each observable depends only on the corresponding state and not any other observable. The current work entails a maximally-interpolated departure from that convention (according to training dataset size) in an attempt to leverage anomalous statistical information in the neighborhood of coding-noncoding transitions (e.g., the exon-intron, introns-exon, junk-exon, or exon-junk transitions, collectively denoted as 'eij-transitions'). The regions of anomalous statistics are often highly structured, having consensus sequences that strongly depart from the strong independence assumptions of the 1st order HMM. The existence of such consensus sequences suggests that we adopt an observation model that has a higher order Markov property with respect to the observations. Furthermore, since the consensus sequences vary by the type of transition, this observational Markov order should be allowed to vary depending on the state.

In the Viterbi context, for a given state dimer transition, such as e_0e_1 or e_0i_0, we can boost the contributions of the corresponding base emissions to the correct prediction of state by using extended states. Specifically, when encountered sequentially in the Viterbi algorithm, the sequence of eij-transition *footprint* states would conceivably score highly when computed for the footprint-width number of footprint-states that overlap the eij-transition (as the generalized clique is moved from left-to-right over the HMM graphical model, as shown in Fig. 1). In other words we can expect a *natural boosting* effect for the correct prediction at such eij-transitions (compared to the standard HMM).

The meta-state, clique-generalized, HMM entails a clique-level factorization rather than the standard HMM factorization (that describes the state transitions with no dependence on local sequence information). This is described in the general formalism to follow, where specific equations are given for application to eukaryotic gene structure identification.

Observation and state dependencies in the generalized-clique HMM are parameterized

independently according to the following.

1) Non-negative integers L and R denoting left and right maximum extents of a substring, w_n, (with suitable truncation at the data boundaries, b_0 and b_{N-1}) are associated with the primitive observation, b_n , in the following way:

$w_n = b_{n-L+1}, \ldots, b_n, \ldots, b_{n+R}$
$\widetilde{w}_n = b_{n-L+1}, \ldots, b_n, \ldots, b_{n+R-1}$

2) Non-negative integers $\boldsymbol{\ell}$ and r are used to denote the left and right extents of the extended (footprint) states, f. Here, we show the relationships among the primitive states λ, dimer states s, and footprint states f:

$s_n = \lambda_n \lambda_{n+1}$ (dimer state, length in λ's =2)
$f_n = s_{n-\boldsymbol{\ell}+1}, \ldots, s_{n+r} \cong \lambda_{n-\boldsymbol{\ell}+1}, \ldots, \lambda_n, \ldots, \lambda_{n+r+1}$ (footprint state, length in s's= $\boldsymbol{\ell}$+r)

As in the 1st order HMM, the nth base observation b_n is aligned with the nth hidden state λ_n:

With the choice of first and last clique described in Fig. 1, we have introduced some additional state and observation primitives (associated with unit-valued transition and emission probabilities) for suitable values of L, R, $\boldsymbol{\ell}$, and r. These additional primitives are shown in Table 1 below.

Table 1 Additional primitives for completion of boundary cliques

Additional Primitives	Type of Primitive	Boundary
$\lambda_{-R-\boldsymbol{\ell}+1}, \ldots, \lambda_{-1}$	States	Left
$b_N, \ldots, b_{N+L+R-2}$	Observations	Right
$\lambda_N, \ldots, \lambda_{N+L+r+1}$	States	Right

Given the above, the clique-factorized HMM is as follows:

$$P(B, \Lambda) = P(w_{-R}, f_{-R}) \{ \boldsymbol{\Pi}_{n=-R+1}^{N+L-2} [P(w_n, f_{n-1}, f_n) / P(\widetilde{w}_n, f_{n-1})] \}$$

A generalization to the Viterbi algorithm can now be directly implemented, using the above form, to establish an efficient dynamic programming table construction. Generalized expressions for the Baum-Welch algorithm are also possible. Some of the generalizations are straightforward extensions of the algorithms from 1st order theory with its minimal clique. Sequence-dependent transition properties in the generalized-clique formalism have no counterpart in the standard 1st Order HMM formalism, however, and that will be elaborated upon here.

The core term in the clique-factorization is:

$$\frac{P(w_n, f_{n-1}, f_n)}{P(\widetilde{w}_n, f_{n-1})} = \frac{P(w_n|f_{n-1}, f_n)\, P(f_n|f_{n-1})\, P(f_{n-1})}{P(\widetilde{w}_n|f_{n-1})\, P(f_{n-1})}$$

In the standard Markov model R = 0, L = 1, r = -1, l = 0: $f_n = \lambda_n$, $w_n = b_n$, $P(\widetilde{w}_n, f_{n-1}) = P(\lambda_n)$:

$$\left.\frac{P(w_n, f_{n-1}, f_n)}{P(\widetilde{w}_n, f_{n-1})}\right|_{\substack{Standard\ Hidden \\ Markov\ Model}} = P(b_n|\lambda_n)\, P(\lambda_n|\lambda_{n-1})$$

In the above we introduce the constraint notation with the vertical bar notation, where the expression on the left is the clique factorization term with the constraint that it approximate according to the standard HMM conditional probabilities.

The core term in the clique-factorization can also be written by introducing a Bayesian parameter, one that happens to provide a matching joint probability construct (to the extent possible) with the term in the numerator:

$$\frac{P(w_n, f_{n-1}, f_n)}{P(\widetilde{w}_n, f_{n-1})} = \frac{P(w_n, f_{n-1}, f_n)}{\sum_{f'_{n(allowed)}} P(\widetilde{w}_n, f_{n-1}, f'_n)} = \frac{P(w_n|f_{n-1}, f_n)\, P(f_n|f_{n-1})\, P(f_{n-1})}{\sum_{f'_n} P(\widetilde{w}_n|f_{n-1}, f'_n)\, P(f'_n|f_{n-1})\, P(f_{n-1})}$$

We now examine specific cases of this equation to clarify the novel improvements that result. ***In what follows we constrain our model to have a minimum length on regions (thus self-transitions) such that footprint states, and their transitions, can only have one transition between different states.***

Consider the case with the first footprint state being of eij-transition type, and the second footprint thereby constrained to be of the appropriate xx-type:

$$\left.\frac{P(w_n, f_{n-1}, f_n)}{P(\widetilde{w}_n, f_{n-1})}\right|_{f_{n-1} \in eij} = \left.\frac{P(w_n, f_{n-1}, f_n)}{\sum_{f'_{n(allowed)}} P(\widetilde{w}_n, f_{n-1}, f'_n)}\right|_{\substack{f_{n-1} \in eij \\ [f'_n unique \in xx]}}$$

$$= P(b_{n+R}|\widetilde{w}_n, f_{n-1}, f_n)\big|_{f_{n-1} \in eij}\, P(f_n|f_{n-1})|_{f_{n-1} \in eij}$$
$$= P(b_{n+R}|\widetilde{w}_n, f_{n-1})$$

Where use is made of the relation $P(f_n|f_{n-1})|_{f_{n-1} \in eij} = 1$ for the unique xx-footprint that follow the eij-transition given our minimum length constraint.

Consider, next, the case with the first footprint state being xx-type:

$$\left.\frac{P(w_n, f_{n-1}, f_n)}{P(\widetilde{w}_n, f_{n-1})}\right|_{f_{n-1} \in xx} = \left.\frac{P(w_n|f_{n-1}, f_n)|_{f_{n-1} \in xx}\, P(f_n|f_{n-1})}{\sum_{f'_n} P(\widetilde{w}_n|f_{n-1}, f'_n)|_{f_{n-1} \in xx}\, P(f'_n|f_{n-1})}\right|_{f_{n-1} \in xx}$$

$$= \left.\frac{P(w_n|f_n)\, P(f_n|f_{n-1})}{\sum_{f'_n} P(\widetilde{w}_n|f'_n) P(f'_n|f_{n-1})}\right|_{f_{n-1} \in xx}$$

If the second footprint is eij-transition type, then the equation has two sum terms in the denominator if the first transition is ii or jj transition, and a third sum contribution (the term with 'f_{ey}') if the first transition is an ee-transition:

In what follows, dimer notation is used on footprints, since we are interested in the footprint-to-footprint transitions. Given their large overlap dependence, this notation and formalism directly generalizes to the same cases no matter the size of the footprint (due to the single major-transition in or between footprints constraint that is provided by a minimum length constraint).

If $f_{n-1} \in xx$ we have three cases: $xx \in \{ii, ee, jj\}$. For $f_{n-1} = ii$, we have two possible $f_n \in \{ii, ie\}$; for $f_{n-1} = jj$, we have two possible $f_n \in \{jj, je\}$; for $f_{n-1} = ee$, we have three possible $f_n \in \{ee, ej, ei\}$.

$$\left.\frac{P(w_n, f_{n-1}, f_n)}{P(\widetilde{w}_n, f_{n-1})}\right|_{\substack{f_{n-1}=ii,\\ f_n=ie}} = \frac{P(w_n|ie)\,P(ie|ii)}{P(\widetilde{w}_n|ie)P(ie|ii)+P(\widetilde{w}_n|ii)P(ii|ii)} = \frac{P(b_{n+R}|\widetilde{w}_n, ie)}{1+\left(\frac{P(\widetilde{w}_n|ii)}{P(\widetilde{w}_n|ie)}\right)\left(\frac{P(ii|ii)}{P(ie|ii)}\right)}$$

Where we have introduced the notation 'ii' to denote the dimer state or the footprint state 'ii…iii', and the notation 'ie' to denote the dimer state or the footprint state 'ii…iie'.

Similarly, consider $f_{n-1} = jj$ and $f_n = je$:

$$\left.\frac{P(w_n, f_{n-1}, f_n)}{P(\widetilde{w}_n, f_{n-1})}\right|_{\substack{f_{n-1}=jj,\\ f_n=je}} = \frac{P(w_n|je)\,P(je|jj)}{P(\widetilde{w}_n|je)P(je|jj)+P(\widetilde{w}_n|jj)P(jj|jj)} = \frac{P(b_{n+R}|\widetilde{w}_n, je)}{1+\left(\frac{P(\widetilde{w}_n|jj)}{P(\widetilde{w}_n|je)}\right)\left(\frac{P(jj|jj)}{P(je|jj)}\right)}$$

For the $f_{n-1} = ee$ and $f_n = ej$ we get a similar expression, but a third term in the sum due to the three possibilities allowed for f_n:

$$\left.\frac{P(w_n, f_{n-1}, f_n)}{P(\widetilde{w}_n, f_{n-1})}\right|_{\substack{f_{n-1}=ee,\\ f_n=ej}} = \frac{P(w_n|ej)\,P(ej|ee)}{P(\widetilde{w}_n|ej)P(ej|ee)+P(\widetilde{w}_n|ei)P(ei|ee)+P(\widetilde{w}_n|ee)P(ee|ee)}$$

$$= \frac{P(b_{n+R}|\widetilde{w}_n, ej)}{1+\left(\frac{P(\widetilde{w}_n|ei)}{P(\widetilde{w}_n|ej)}\right)\left(\frac{P(ei|ee)}{P(ej|ee)}\right)+\left(\frac{P(\widetilde{w}_n|ee)}{P(\widetilde{w}_n|ej)}\right)\left(\frac{P(ee|ee)}{P(ej|ee)}\right)}$$

Likewise for the $f_{n-1} = ee$ and $f_n = ei$ we get a similar expression, but a third term in the sum:

$$\left.\frac{P(w_n, f_{n-1}, f_n)}{P(\widetilde{w}_n, f_{n-1})}\right|_{\substack{f_{n-1}=ee,\\ f_n=ei}} = \frac{P(w_n|ei)\,P(ei|ee)}{P(\widetilde{w}_n|ei)P(ei|ee)+P(\widetilde{w}_n|ej)P(ej|ee)+P(\widetilde{w}_n|ee)P(ee|ee)}$$

$$= \frac{P(b_{n+R}|\widetilde{w}_n, ei)}{1+\left(\frac{P(\widetilde{w}_n|ej)}{P(\widetilde{w}_n|ei)}\right)\left(\frac{P(ej|ee)}{P(ei|ee)}\right)+\left(\frac{P(\widetilde{w}_n|ee)}{P(\widetilde{w}_n|ei)}\right)\left(\frac{P(ee|ee)}{P(ei|ee)}\right)}$$

Consider now the cases involving self-transitions: $f_{n-1} = xx$ and $f_n = xx$. The derivation parallels that above for $f_{n-1} = ii$ and $f_n = ii$:

$$\left.\frac{P(w_n, f_{n-1}, f_n)}{P(\widetilde{w}_n, f_{n-1})}\right|_{\substack{f_{n-1}=ii,\\ f_n=ii}} = \frac{P(w_n|ii)\,P(ii|ii)}{P(\widetilde{w}_n|ie)P(ie|ii)+P(\widetilde{w}_n|ii)P(ii|ii)} = \frac{P(b_{n+R}|\widetilde{w}_n, ii)}{1+\left(\frac{P(\widetilde{w}_n|ie)}{P(\widetilde{w}_n|ii)}\right)\left(\frac{P(ie|ii)}{P(ii|ii)}\right)}$$

Similarly, consider $f_{n-1} = jj$ and $f_n = jj$:

$$\frac{P(w_n,f_{n-1},f_n)}{P(\widetilde{w}_n,f_{n-1})}\Big|_{\substack{f_{n-1}=jj,\\ f_n=jj}} = \frac{P(w_n|jj)\,P(jj|jj)}{P(\widetilde{w}_n|je)P(je|jj)+P(\widetilde{w}_n|jj)P(jj|jj)} = \frac{P(b_{n+R}|\widetilde{w}_n,jj)}{1+\left(\frac{P(\widetilde{w}_n|je)}{P(\widetilde{w}_n|jj)}\right)\left(\frac{P(je|jj)}{P(jj|jj)}\right)}$$

For the $f_{n-1} = ee$ and $f_n = ej$ we get the third term in the sum due to the three possibilities allowed for f_n:

$$\frac{P(w_n,f_{n-1},f_n)}{P(\widetilde{w}_n,f_{n-1})}\Big|_{\substack{f_{n-1}=ee,\\ f_n=ee}} = \frac{P(w_n|ee)\,P(ee|ee)}{P(\widetilde{w}_n|ej)P(ej|ee)+P(\widetilde{w}_n|ei)P(ei|ee)+P(\widetilde{w}_n|ee)P(ee|ee)}$$

$$= \frac{P(b_{n+R}|\widetilde{w}_n,ee)}{1+\left(\frac{P(\widetilde{w}_n|ei)}{P(\widetilde{w}_n|ee)}\right)\left(\frac{P(ei|ee)}{P(ee|ee)}\right)+\left(\frac{P(\widetilde{w}_n|ej)}{P(\widetilde{w}_n|ee)}\right)\left(\frac{P(ej|ee)}{P(ee|ee)}\right)}$$

In the above expressions we clearly have sequence dependent transitions. For $f_{n-1} = ii,\ and\ f_n = ie$ for example, we have:

$$\rho|_{GCHMM} = \frac{P(w_n,f_{n-1},f_n)}{P(\widetilde{w}_n,f_{n-1})}\Big|_{\substack{f_{n-1}=ii,\\ f_n=ie}} = \frac{P(w_n|ie)\,P(ie|ii)}{P(\widetilde{w}_n|ie)P(ie|ii)+P(\widetilde{w}_n|ii)P(ii|ii)} = \frac{P(b_{n+R}|\widetilde{w}_n,ie)P(ie|ii)}{P(ie|ii)+P(ii|ii)\left(\frac{P(\widetilde{w}_n|ii)}{P(\widetilde{w}_n|ie)}\right)}$$

While the standard HMM has this ratio with w_n a single element emission sequence, and $P(w_n, f_{n-1}, f_n) = P(w_n|f_n)\,P(f_n|f_{n-1})$, thus, for the standard HMM:

$$\rho|_{Std.HMM} = \frac{P(w_n,f_{n-1},f_n)}{P(\widetilde{w}_n,f_{n-1})}\Big|_{\substack{f_{n-1}=ii,\\ f_n=ie,\\ Std.HMM}} = P(b_{n+R}|ie)\,P(ie|ii)$$

If we generalized the Std. HMM to higher order Markov models on emissions, to the same order as in the generalized clique, there is still the difference in the transition probability contributions:

$$\rho|_{\substack{Std.HMM\\ HO\ EMs}} = P(b_{n+R}|\widetilde{w}_n, ie)\,P(ie|ii),$$

as can be seen in the ratio of their contributions, and how it is sequence dependent (i.e., dependent on '$\widetilde{w}_i$'):

$$\frac{\rho|_{\substack{Std.HMM\\ HO\ EMs}}}{\rho|_{GCHMM}} = P(ie|ii) + P(ii|ii)\left(\frac{P(\widetilde{w}_i|ii)}{P(\widetilde{w}_i|ie)}\right).$$

Note that the sequence dependencies (in this and the other footprint transition choices) enter via likelihood ratio terms. These are precisely the type of terms examined in [17] in an effort to improve the HMM-based discriminatory ability via use of SVMs. The 'discriminatory' aspect of the key new (sequence-dependent) contribution is most evident in forms like that above, where we have a likelihood ratio for the observed sequences given the different label 'classifications' chosen. In the cases that follow we will examine the extreme cases of the likelihood-ratio discriminator strongly classifying one way or the other, or not strongly classifying either way with the given sequence information (making the contribution of knowing that sequence

information negligible, which should then reduce to the std. HMM situation, as will be shown). Specifically, we will now examine the above equations in situations where the sequence-dependent likelihood-ratios strongly favor one state model over another, with particular attention as to whether there are sequence dependent scenarios offering recovery of the heavy-tail distribution in example one and recovery of contrast resolution in example two:

Example One:

For $f_{n-1} = ii$ and $f_n = ii$ we showed:

$$\rho = \frac{P(w_n, f_{n-1}, f_n)}{P(\widetilde{w}_n, f_{n-1})}\Bigg|_{\substack{f_{n-1}=ii,\\ f_n=ii}} = \frac{P(b_{n+R}|\widetilde{w}_n, ii)}{1 + \left(\frac{P(\widetilde{w}_n|ie)}{P(\widetilde{w}_n|ii)}\right)\left(\frac{P(ie|ii)}{P(ii|ii)}\right)}$$

Example One; Case 1: $P(\widetilde{w}_n|ie) \cong P(\widetilde{w}_n|ii)$ (likelihood ratio of probabilities is weakly classified)

$$\rho|_{ie\cong ii} \cong P(b_{n+R}|\widetilde{w}_n, ii)\, P(ii|ii)/[P(ii|ii) + P(ie|ii)]$$
$$= P(b_{n+R}|\widetilde{w}_n, ii)\, P(ii|ii)$$

Thus, in the 'uninformed' case we recover regular 1st order HMM theory, with geometric distribution on 'ii'. In this notation, $\rho|_{ie\cong ii}$ refers to the value of ρ when the observed sequence $\widetilde{w}_n$ has approximately the same probability regardless of the state being 'ii' or 'ie'.

Example One; Case 2: $P(\widetilde{w}_n|ie) \gg P(\widetilde{w}_n|ii)$ (likelihood ratio of probabilities is strongly classified)

$$\rho|_{ie\gg ii} \cong P(b_{n+R}|\widetilde{w}_n, ii)\left[\frac{P(\widetilde{w}_n|ii)P(ii|ii)}{P(\widetilde{w}_n|ie)P(ie|ii)}\right]$$

In this case we obtain contributions less than the regular 1st order HMM counterpart, effectively shortening the geometric distribution on 'ii'→ e.g., it adaptively switches to a shorter, sharper, fall-off on the distribution in a sequence dependent manner.

Example One; Case 3: $P(\widetilde{w}_n|ie) \ll P(\widetilde{w}_n|ii)$ (likelihood ratio of probabilities is strongly classified – the other way)

$$\rho|_{ie\ll ii} \cong P(b_{n+R}|\widetilde{w}_n, ii)\ \underline{1}$$

In this case we obtain contributions greater than the regular 1st order HMM theory. In particular, ***we recover the heavy tail distribution in a sequence dependent manner***:

$$\frac{P(w_n, f_{n-1}, f_n)}{P(\widetilde{w}_n, f_{n-1})}\Bigg|_{\substack{f_{i-1}\in ie,\\ f_i\in ee}} = P(b_{n+R}|\widetilde{w}_n, f_{n-1})$$

Example Two:

One more example-case will be considered, that involving acceptor splice-site recognition. For $f_{n-1} = ii, f_n = ie$ we have:

$$\rho = \frac{P(w_n, f_{n-1}, f_n)}{P(\widetilde{w}_n, f_{n-1})}\Big|_{\substack{f_{n-1}=ii,\\ f_n=ie}} = \frac{P(b_{n+R}|\widetilde{w}_n, ie)}{1+\left(\frac{P(\widetilde{w}_n|ii)}{P(\widetilde{w}_n|ie)}\right)\left(\frac{P(ii|ii)}{P(ie|ii)}\right)}$$

Example Two; Case 1: $P(\widetilde{w}_n|ie) \cong P(\widetilde{w}_n|ii)$

$$\rho|_{ie\cong ii} \cong P(b_{n+R}|\widetilde{w}_n, ie)\, P(ie|ii)$$

We recover regular HMM theory in the uninformed situation.

Example Two; Case 2: $P(\widetilde{w}_n|ie) \gg P(\widetilde{w}_n|ii)$

$$\rho|_{ie\gg ii} \cong P(b_{n+R}|\widetilde{w}_n, ie)$$

Greater than regular 1st order HMM theory. Removes key penalty of *P(ie|ii)* factor when sequence match overrides. ***Resolves weak contrast resolution at 1st order.***

Example Two; Case 3: $P(\widetilde{w}_n|ie) \ll P(\widetilde{w}_n|ii)$

$$\rho|_{ie\ll ii} \cong P(b_{n+R}|\widetilde{w}_n, ie)\left[\frac{P(ie|ii)P(\widetilde{w}_n|ie)}{P(ii|ii)P(\widetilde{w}_n|ii)}\right]$$

Less than regular 1st Order HMM, effectively weakens ie transition strength (the classic major-transition bias factor).

The clique factorization also allows for an alternate representation such that the internal scalar-based state discriminant can be replaced with a vector-based feature. This would allow the substitution of a discriminant based on a Support Vector Machine (SVM) as demonstrated for splice sites in [17]. Also, we note that these alternate representations would not introduce any significant increase in computational complexity, since the SVM-based discriminant, having been trained offline, would require the computation of a simple vector dot product. Thus, the likelihood ratio look-up can simply be to the tabulated sequence probability estimates (based on counts, as outlined in what follows), or make use of BLAST (homology-based) test, or an SVM-based test (the latter two cases areas of ongoing work, see Discussion).

(4) Enumeration of the Footprint States

According to the restrictions just described, footprint states fall into the same two categories or types as dimer states, xx-type and eij-type. Regardless of footprint state type, each footprint state can be considered to be generated by the xx-type dimer that it contains. For xx-types, it is sufficient to specify the generating dimer only, such as $\mathtt{i}_0\mathtt{i}_0$ for the xx-type footprint state $\mathtt{i}_0\mathtt{i}_0\ldots\mathtt{i}_0$. For eij-types, a position must also be specified for the location of the generating

dimer within the generated footprint state. The number of xx-type footprint states is identical to the number of xx-type dimers, as enumerated in Table 1 below.

Dimer Index	XX- type Generating Dimer	XX- type Generated Footprint State
0	i_0i_0	$i_0i_0\ldots i_0$
1	i_1i_1	$i_1i_1\ldots i_1$
2	i_2i_2	$i_2i_2\ldots i_2$
3	$\hat{i}_0\hat{i}_0$	$\hat{i}_0\hat{i}_0\ldots\hat{i}_0$
4	$\hat{i}_1\hat{i}_1$	$\hat{i}_1\hat{i}_1\ldots\hat{i}_1$
5	$\hat{i}_2\hat{i}_2$	$\hat{i}_2\hat{i}_2\ldots\hat{i}_2$
6	e_0e_1	$e_0e_1\ldots e_{(F)mod3}$
7	e_1e_2	$e_1e_2\ldots e_{(F+1)mod3}$
8	e_2e_0	$e_2e_0\ldots e_{(F-1)mod3}$
9	$\hat{e}_0\hat{e}_1$	$\hat{e}_0\hat{e}_1\ldots\hat{e}_{(F)mod3}$
10	$\hat{e}_1\hat{e}_2$	$\hat{e}_1\hat{e}_2\ldots\hat{e}_{(F+1)mod3}$
11	$\hat{e}_2\hat{e}_0$	$\hat{e}_2\hat{e}_0\ldots\hat{e}_{(F-1)mod3}$
12	jj	$jj\ldots j$

Table 1. All 13 xx-type footprint states generated by the xx-type dimmers

As for the eij-type footprint states, each is generated by the non-homogeneous dimer that it contains but is further characterized by the position of the generating dimer within the footprint string, such as e_0i_0 in the right-most position of the eij-type footprint state $e_{(F-2)mod3}e_{(F-1)mod3}\ldots e_0e_0e_0i_0$. As a consequence of this, there are F eij-type footprint states for each corresponding eij-type dimer. Given an eij-type footprint state of length F in dimers, there are precisely F possible positions for the implied eij-type dimer to occur within the footprint state's string of primitives. These dimer positions are labeled 0, …, F-1 and taken in the order of encoding (forward or reverse) in Table 2 below. Thus we have the relation: # eij-type footprint states= 20 (F) = (# eij-type dimer states) (F).

Dimer Index	EIJ-type Generating Dimer	EIJ-type Generated Footprint State For Generating Dimer Positions 0, …, F-1		
		0	…	F-1
0	e_0i_0	$e_0i_0\ldots i_0$	…	$e_{(1-F)mod3}e_{(2-F)mod3}\ldots e_0i_0$
1	e_1i_1	$e_1i_1\ldots i_1$	…	$e_{(2-F)mod3}e_{(-F)mod3}\ldots e_1i_1$
2	e_2i_2	$e_2i_2\ldots i_2$	…	$e_{(-F)mod3}e_{(1-F)mod3}\ldots e_2i_2$
3	$\hat{e}_0\hat{i}_0$	$\hat{e}_{(1-F)mod3}\hat{e}_{(2-F)mod3}\ldots\hat{e}_0\hat{i}_0$	…	$\hat{e}_0\hat{i}_0\ldots\hat{i}_0$
4	$\hat{e}_1\hat{i}_1$	$\hat{e}_{(2-F)mod3}\hat{e}_{(-F)mod3}\ldots\hat{e}_1\hat{i}_1$	…	$\hat{e}_1\hat{i}_1\ldots\hat{i}_1$
5	$\hat{e}_2\hat{i}_2$	$\hat{e}_{(-F)mod3}\hat{e}_{(1-F)mod3}\ldots\hat{e}_2\hat{i}_2$	…	$\hat{e}_2\hat{i}_2\ldots\hat{i}_2$
6	i_0e_1	$i_0e_1e_2\ldots e_{(F)mod3}$	…	$i_0\ldots i_0e_1$
7	i_1e_2	$i_1e_2e_0\ldots e_{(F+1)mod3}$	…	$i_1\ldots i_1e_2$
8	i_2e_0	$i_2e_0e_1\ldots e_{(F-1)mod3}$	…	$i_2\ldots i_2e_0$
9	$\hat{i}_0\hat{e}_1$	$\hat{i}_0\ldots\hat{i}_0\hat{e}_1$	…	$\hat{i}_0\hat{e}_1\hat{e}_2\ldots\hat{e}_{(F)mod3}$
10	$\hat{i}_1\hat{e}_2$	$\hat{i}_1\ldots\hat{i}_1\hat{e}_2$	…	$\hat{i}_1\hat{e}_2\hat{e}_0\ldots\hat{e}_{(F+1)mod3}$
11	$\hat{i}_2\hat{e}_0$	$\hat{i}_2\ldots\hat{i}_2\hat{e}_0$	…	$\hat{i}_2\hat{e}_0\hat{e}_1\ldots\hat{e}_{(F-1)mod3}$
12	$(e_2j)_{TAA}$	$(e_2j)_{TAA}jj\ldots j$	…	$e_{(-F)mod3}e_{(1-F)mod3}\ldots(e_2j)_{TAA}$
13	$(e_2j)_{TAG}$	(Similar to above)	…	(Similar to above)
14	$(e_2j)_{TGA}$	" "	…	" "
15	$(\hat{e}_2j)_{TAA}$	$\hat{e}_{(-F)mod3}\hat{e}_{(1-F)mod3}\ldots(\hat{e}_2j)_{TAA}$	…	$(\hat{e}_2j)_{TAA}j\ldots j$
16	$(\hat{e}_2j)_{TAG}$	(Similar to above)	…	(Similar to above)

17	$(\hat{e}_2 j)_{TGA}$	" "	…	" "
18	je_0	$je_0e_1\ldots e_{(F-1)mod3}$	…	$jj\ldots je_0$
19	$j\hat{e}_0$	$jj\ldots j\hat{e}_0$	…	$j\hat{e}_0\hat{e}_1\ldots\hat{e}_{(F-1)mod3}$

Table 2. All 20(F) eij-type footprint states generated by the eij-type dimers

We have the following relations:

footprint states = 13 + 20(F)
footprint state transitions = 13 + 20(F+1)

In the model without the minimum length constraint we still have the fundamental set of 33 dimers, beyond that, however, the larger footprints can have arbitrary numbers of state-toggles:

extended states without minimum length assumption $\geq 33 * 2^{F-1}$
extended state transitions without minimum length assumption $\geq 33 * 2^{F}$

(5) Measures of Predictive Performance that are used

The measure of prediction performance was taken in two ways: full exon accuracy and individual base (nucleotide) accuracy, according to the conventions of Burset and Guigo in [16].

Accuracy at the base or nucleotide level, is given by

snsp_avg = (sn + sp*)/2, where sn = TP/(TP + FN) and sp* = TP/(TP + FP), and

TP = true positives; FP = false positives; FN = false negatives

Note that the authors [16] have used an alternative form of specificity from the usual form

sp=TN/(TN+FP).

This is done in the context of gene prediction, with typically high concentrations of junk, where the contribution from the quantity TN= true negative (or correctly predicted actual non-coding) can overwhelm FP in what is actually weakly accurate prediction (i.e., scoring is best conveyed in terms of the overlap between predicted positives and actual positives [36]).

We use (sn + sp*)/2 for accuracy, following the conventions of [16], partly to compare with their results, but we also calculate the specificity according to the standard form sp=TN/(TN+FP), and both of these values are shown in Tables 8 & 9. The specificity convention sp* = TP/(TP + FP) has the effect of weighting genes with shorter and fewer exons more heavily in the base and exon level accuracy measurements, respectively. (In the notation to follow, sp will be used in place of sp* if there is no ambiguity.) Moreover, this effect can become extremely pronounced in cases such as both of the cited evaluations, where all DNA sequences tested contain only a *single gene*. In this effort, the number of correct (and incorrect) predictions are first summed over all test sequences and then the measurements were computed from those sums for the exon and base level measurements, respectively. Either method of measurement appears appropriate for the Burset and Guigo data sets, where the data sequences have a single gene via pre-processing (and may be *leveraged* as such in the design of the program being tested). In what is a more realistic context of raw genomic data processing, however, we are likely to encounter two key issues as part of the problem:

1) We have raw genomic sequences that contain multiple genes.

2) Scoring at the exon level in effect designates the *exon* as the fundamental unit being counted rather than the *gene*, this avoids weighing more complex genes the same as simpler genes (that have fewer exons).

As indicated above, in each case of the data sets used in this effort, the measurements for both the exon and base level prediction differ somewhat from the method used in the cited evaluations. Moreover, of the data sets tested in this effort, ALLSEQ is the only data set consisting entirely of single-gene DNA sequences. The results of the meta-state HMM for ALLSEQ in this effort are given in both the cited measure of accuracy [6], as well as standard 'exon-level' scoring.

The accuracy measure at the full exon level presents a much greater challenge as it requires the successful prediction of the entire exon for the exon to be scored as correct. These events include the start and end positions of exons as well as the continuation of the exon at all intermediate introns splicing points. The full exon accuracy is given similarly to that given before at base-level scoring:

$$SNSP\_AVG = (SN + SP^*)/2, \text{ where}$$
$$SN = (\text{number of correct exons})/(\text{number of actual exons}), \text{ and}$$
$$SP^* = (\text{number of correct exons})/(\text{number of predicted exons})$$

Again, SP will be used in place of SP* in what follows if there is no ambiguity. It should be noted that this measure for full exon accuracy does not allow for any improvement due to *partial* exon prediction. More specifically, the exon level accuracy can only be improved by the precise prediction of one or more *entire* exons – at both start and end positions.

Results

All predictions are based on state prior, state transition, and emission probabilities which are estimated directly from counts in the training data without any further refinement. The meta-state HMM model is interpolated to highest Markov order on emission probabilities given the training data size, and to highest Markov order (subsequence length) on the footprint states (with different values shown in the Results as multi-trajectory plots). The former is accomplished via simple count cutoff rules, the latter via an identification of anomalous base statistics near the coding/noncoding-transitions, initially, followed by direct HMM performance tuning. Allowed footprint transitions are restricted to those that have at most one coding/noncoding-transition, which leads to only linear growth in state number with footprint size, *not geometric growth*, enabling the full advantage of generalized-clique modeling at a computational expense little more than that of a standard HMM.

Algorithmic Complexity of meta-HMM dynamic programming table construction

For comparison with the meta-state HMM, we first consider the complexity of the traditional 1st order HMM. First define 'T' as the length of the testing data set, and 'N' as the number of states. The Viterbi algorithm constructs the table recursively, with computational updates in each cell in a given column only dependent on computations involving each of the cells of the prior column, thus the time complexity involved in the Viterbi algorithm is given by $O(TN^2)$. In the meta-state HMM we have similar growth in number of states, but in the case of the increasing footprint size F this increase in states, *and state transitions*, is linear, with time complexity given by $O(T(F+L+R))$, where linearity in F for fixed L and R is verified in the set of time trials shown in Fig. 2.

Results for Benchmark Dataset ALLSEQ
Exon- and base-level accuracy for values of the parameters M, F, L, and R were tested and examined for stability. Fig. 3 and Fig. 4 below show plots for exon- and base-level maxima, respectively, over the parameters L and R of meta-state HMM's prediction performance. The plots illustrate the enhanced performance of the meta-state HMM over simpler prediction models, including the (null hypothesis result) meta-state HMM for which the base Markov parameter, M=0. (Note: the meta-state HMM uses only the *intrinsic* information in the data – making no use of *extrinsic* information, such as EST's, protein homology, etc.)

In comparing the results of this data set to the other results in this effort, the quality of the best result can be attributed to the increased size of the training data set (despite the decreased coding density) as well as adherence among the donor and acceptor splice sites to the consensus sequences, gt and ag, respectively. Fig.s 3 & 4 also show the best performing predictors from the original benchmark study, FGENEH and GeneID+, that use intrinsic and extrinsic genomic information, respectively. At both the full exon- and base- levels, the meta-state HMM outperforms standard HMM approaches by a discernable margin.

Results for C. elegans Dataset
The results shown in Fig.'s 5 & 6 indicate that a local maximum for the exon and base level predictions was attained at F=12, with a plateau for F>12 extending to F=20, with exact exon prediction accuracy 74% and base accuracy 90%. In comparing the results of this data set to the other results in this effort, the reduced performance at full exon level for M=8 compared to that for M=5 is an indication of insufficient training size reflected in lack of support for M=8 probability estimates at splice sites.

The degree of preconditioning in our data set is minimal, such that there is allowance in the data for disagreement with the consensus dinucleotide introns sequences, gt and ag, as well as the incorporation of reverse encodings. As mentioned previously, we arrive at a base accuracy of 90%. The prospects for improving this result further are many, starting with simply enlarging the training dataset by including similar genomes from other nematodes, *C. Briggsiae* in particular (see further discussion in the Conclusion).

Discussion
The top performing results from the evaluations performed in [16] and [14] are included in Table 6 & 7 below (where they predict on data that has much greater preprocessing, not raw genome), including values for the (nucleotide) base level accuracy converted from the AC measurement to E[(sn+sp)/2].

Table 8 shows the top results of the meta-state HMM for the data sets and parameter values tested in this effort, including in each case the optimum values for the parameters M, F, L and R. Recall that the method of measurement used in this effort differs slightly from that of the cited evaluations. For additional reference, Table 9 shows the maximum accuracy specifically for the ALLSEQ data set at both the base and exon levels using the method of measurement in the cited evaluations, as well as our own.

Software	Nucleotide level	Full Exon Level

Name	E[sn]	E[sp]	AC	E[(sn+sp)/2]	E[SN]	E[SP]	E[(SN+SP)/2]
FGENEH	0.77	0.88	0.78±0.26	0.825	0.61	0.64	0.64±0.33
GeneID+	0.91	0.91	0.88±0.16	0.91	0.73	0.70	0.71±0.29

Table 6. Top 2 performers in the evaluation by Burset and Guigo testing with **ALLSEQ**.

Software Name	Nucleotide level				Full Exon Level		
	E[sn]	E[sp]	AC	E[(sn+sp)/2]	E[SN]	E[SP]	E[(SN+SP)/2]
Genie	0.91	0.90	0.89±0.16	0.905	0.71	0.70	0.71±0.30
Genscan	0.95	0.90	0.91±0.12	0.925	0.70	0.70	0.70±0.32
HMMgene	0.93	0.93	0.91±0.13	0.93	0.76	0.77	0.76±0.30

Table 7. Top 3 performers in the evaluation by Rogic, et al., testing with HMR195.

The meta-state HMM's performance on the ALLSEQ dataset clearly exceeds that of the top performing program, GeneID+, cited in [16], by substantial margins, 6.5% and 17%, at the base- and exon-levels, respectively. GeneID+ also uses *extrinsic* information via "amino acid similarity searches" in the process of forming its prediction, whereas the meta-state HMM in this effort uses only the *intrinsic* information contained in the DNA sequence data alone.

Data set Name	Nucleotide level					Full Exon Level				
	sn	sp	(sn+sp)/2	M	F	SN	SP	(SN+SP)/2	M	F
ALLSEQ	0.978	0.954	0.966	8	4	0.919	0.803	0.861	8	12
Chr. I-V	0.938	0.864	0.901	5	12	0.775	0.711	0.743	2	20

Table 8. Maximum accuracy of meta-state HMM for the parameter values tested.

Data set Name	Nucleotide level					Full Exon Level				
	E[sn]	E[sp]	E[(sn+sp)/2]	M	F	E[SN]	E[SP]	E[(SN+SP)/2]	M	F
ALLSEQ	0.987	0.961	0.974	8	12	0.917	0.847	0.882	8	12

Table 9. Maximum accuracy of meta-state HMM for ALLSEQ using the cited method of measurement

The question naturally arises on how we might do better, and we are proceeding in three directions: (1) verifying that HMMD offers little improvements due to the recovery of the heavy tail attribute see [24]; (2) future work involving pMM/SVM sensors [17]; (3) future work involving alternative-splice state structures [43] (with verification of statistical support for the more elaborate state model indicated in [27]); and (4) use of large footprints of HMMD scaffolding to employ zone-dependent statistics to capture *cis*-regulatory signaling, in particular, in the generalized meta-HMMD model. In this effort we tried to mainly draw comparisons with other methods similarly based solely on intrinsic genomic statistics. The method presented here will benefit from extrinsic genomic information 'add-ons' for boosting performance via use of homology matching, or EST alignment, for example. We do not compare with the state-of-the-art extrinsic/intrinsic techniques in this purely intrinsic approach, but upon the further extrinsic/intrinsic statistical modeling refinements indicated above, such a comparison will be

made and judging from the performance of the meta-HMM modeling foundation, a state-of-the-art gene structure identifier should result.

Conclusion

We describe a clique-generalized, meta-state, HMM. The model involves both observations and states of extended length in a generalized clique structure, where the extents of the observations and states are incorporated as parameters in the new model. This clique structure was intended to address the following 2-fold hypothesis.

1) The introduction of extended observations would take greater advantage of the information contained in higher order, position-dependent, signal statistics in DNA sequence data taken from extended regions surrounding coding/noncodong sites; and
2) The introduction of extended states would attain a natural boosting by repeated look-up of the tabulated statistics associated in each case with the given type of coding/non-coding boundary.

We find that our meta-state HMM approach enables a stronger HMM-based framework for the identification of complex structure in stochastic sequential data. We show an application of the meta-state HMM to the identification of eukaryotic gene structure in the *C. elegans* genome. We have shown that the performance of the meta-state HMM-based gene-finder performs comparably to three of the best gene-finders in use today, GENIE, GENSCAN and HMMgene [44]. The method shown here, however, is the bare-bones HMM implementation without use of signal sensors to strengthen localized encoding information, such as splice site information. An SVM-based improvement, to integrate directly with the approach introduced here, is described in [17], and given the successful use of neural-net discriminators to improve splice-site recognition in the GENIE gene finder [45], there are clear prospects for further improvement in overall gene-finding accuracy with the meta-state HMM foundation described in this paper.

Acknowledgments

Author SWH would like to thank Ph.D. Advisor David Haussler for originally posing the question of ‘what might be gained by use of higher-order state models?’ (in a graduate bioinformatics course at UCSC in 1999). Finding the answer has taken a bit longer than expected. Both authors would also like to thank Meta Logos Inc. for allowing academic research with unrestricted use of the clique-generalized HMM software tools developed by SWH while co-Founding Meta Logos.

Figures

Figure 1. Top Panel. Sliding-window association (clique) of observations and hidden states in the meta-state hidden Markov model, where the clique-generalized HMM algorithm describes a left-to-right traversal (as is typical) of the HMM graphical model with the specified clique window. The first observation, b_0, is included at the leading edge of the clique overlap at the HMM's left boundary. For the last clique's window overlap we choose the trailing edge to include the last observation b_{N-1}. Bottom Panel. Graphical model of the clique-generalized HMM, where the interconnectedness on full joint dependencies is only partly drawn. The graphical model is significantly constrained, as well, in a manner not represented in the graphical model, in that state sequences are only allowed with at most one non-self transition.

Figure 2. Meta-state HMM test times for test data length 1Mb

Figure 3. Maximum full exon meta-state HMM performance for data ALLSEQ

Figure 4. Maximum base level meta-state HMM performance for data ALLSEQ

Figure 5, F-view. **Top.** Full exon level accuracy for *C. elegans* with 5-fold cross-validation. **Bottom.** Base level accuracy for *C. elegans* with 5-fold cross-validation.

Figure 6, M-view. **Top.** Full exon level accuracy for *C. elegans* 5-fold cross-validation. **Bottom.** Base level accuracy for *C. elegans* 5-fold cross-validation.

Bibliography

1. Stanke M. and B. Morgenstern. AUGUSTUS: a web server for gene prediction in eukaryotes that allows user-defined constraints. Nucleic Acids Research, 2005, Vol. 33, W465–W467

2. Rajapakse, J. C. and L. S. Ho. Markov Encoding for Detecting Signals in Genomic Sequences. IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 2, No. 2,, pgs. 131-142.

3. Majoros, W.H., M. Pertea and S. L. Salzberg . TigrScan and GlimmerHMM: two open source *ab initio* eukaryotic gene-finders. Bioinformatics, Vol. 20 no. 16 2004, pages 2878–2879.

4. Taher, L., O. Rinner, S. Garg, A. Sczyrba, M. Brudno, S. Batzoglou and B. Morgenstern. AGenDA: homology-based gene prediction. Bioinformatics Vol. 19 no. 12 2003, pages 1575–1577.

5. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. *J Mol Biol* 215 (3): 403–410. 1990.

6. Sonnenburg S., A. Zien, and G. Ratsch . ARTS: accurate recognition of transcription starts in human. Bioinformatics Vol. 22 no. 14 2006, pages e472–e480.

7. Do J.H. and D-K. Choi. Computational Approaches to Gene Prediction . The Journal of Microbiology, April 2006, Vol. 44, No. 2. p.137-144

8. Korf I. Gene finding in novel genomes. BMC Bioinformatics 2004, 5:59.

9. Mathe C., M.-F. Sagot, T. Schiex and P. Rouze . Current methods of gene prediction, their strengths and weaknesses. Nucleic Acids Research, 2002, Vol. 30 No. 19, 4103-4117.

10. Allen, J. E., W. H. Majoros, M. Pertea and S. L. Salzberg. JIGSAW, GeneZilla, and GlimmerHMM: puzzling out the features of human genes in the ENCODE regions. Genome Biology 2006, 7(Suppl 1):S9

11. Noguchi, H., Park, J., Takagi, T.: MetaGene: prokaryotic gene finding from environmental genome shotgun sequences. Nucleic Acids Res. 2006 , 34 (19) :5623-30

12. Taher L., O. Rinner, S. Garg, A. Sczyrba, M. Brudno, S. Batzoglou, and B. Morgenstern. Agenda: homology-based gene prediction. *Bioinformatics*, 19(12):1575-1577, Aug 2003.

13. van Baren MJ, Koebbe BC, Brent MR. Using N-SCAN or TWINSCAN to predict gene structures in genomic DNA sequences. Curr Protoc Bioinformatics. 2007 Dec;Chapter 4:Unit 4.8.

14. Sanja Rogic, Alan K Mackworth, and B.F. Francis Ouellette, "Evaluation of Gene-Finding Programs on Mammalian Sequences," Genome Res. 2001. 11: 817-832, pp. 817-832, 2001.

15. Dunham I., Shimizu N., Roe B.A. & Chissoe S. (1999) The DNA sequence of human chromosome 22. Nature 402, 489-95.

16. Moises Burset and Roderic Guigo, "Evaluation of Gene Structure Prediction Programs," Genomics, vol. 34, pp. 353-367, 1996.

17. Stephen Winters-Hilt and Brian Roux, "Hybrid MM/SVM structural sensors for stochastic sequential data," in Proceedings of the Fifth Annual MCBIOS Conference. Systems Biology: Bridging the Omics. BMC Bioinformatics 2008, 9(Suppl 9):S12.

18. Liu H. , H. Han, J. Li and L. Wong. DNAFSMiner: A Web-Based Software Toolbox to Recognize Two Types of Functional Sites in DNA Sequences. http://sdmc.i2r.a-star.edu.sg/DNAFSMiner/.

19. Sonnenburg S., G. Schweikert, P. Philips, J. Behr and G. Rätsch . Accurate splice site prediction using support vector machines. BMC Bioinformatics 2007, 8(Suppl 10):S7

20. Degroeve, S., Y. Saeys, B. De Baets, P. Rouzé and Y. Van de Peer. SpliceMachine: predicting splice sites from high-dimensional local context representations. Bioinformatics, Vol. 21 no. 8 2005, pages 1332–1338

21. Muro, E.M., R. Herrington, S. Janmohamed, C. Frelin, M. A. Andrade-Navarro and N. N. Iscove. Identification of gene 3' ends by automated EST cluster analysis. PNAS, December 23, 2008, vol. 105, no. 51, pgs. 20286–20290.

22. Bellora N., D. Farre and M. Mar Alba. PEAKS: identification of regulatory motifs by their position in DNA sequences. Bioinformatics, Vol. 23 no. 2 2007, pages 243–244.

23. X. He, X. Ling, and S. Sinha. Alignment and Prediction of cis-Regulatory Modules Based on a Probabilistic Model of Evolution. PLoS Computational Biology 2009,Volume 5, Issue 3, Pgs 1-14.

24. Winters-Hilt S. and Z. Jiang, Hidden Markov model with duration side-information for novel HMMD derivation, with application to eukaryotic gene finding. Submitted to EURASIP Genome signal Processing.

25. Winters-Hilt, S and C Baribault. A novel, fast, HMM-with-Duration implementation – for application with a new, pattern recognition informed, nanopore detector. BMC Bioinf. 8 S7, S19 (2007).

26. Winters-Hilt S. and Jiang Z. A hidden Markov model with binned duration algorithm. IEEE Trans. on Sig Proc., Feb. 2010, Vol 58, Num. 2, pgs 948-952.

27. Winters-Hilt S: Hidden Markov Model Variants and their Application. BMC Bioinformatics 2006, 7 S2: S14.

28. Lu, D. Motif Finding. UNO MS thesis in CS, Summer 2009, Advisor – Prof. S. Winters-Hilt.

29. Shinozaki D., T. Akutsu and O. Maruyama. Finding optimal degenerate patterns in DNA sequences. Bioinformatics Vol. 19 Suppl. 2 2003, pages ii206–ii214
30. Frickey T. and G. Weiller. Mclip: motif detection based on cliques of gapped local profile-to-profile alignments. Bioinformatics, Vol. 23 no. 4 2007, pages 502–503
31. de Hoon, M.J.L., S. Imoto, J. Nolan and S. Miyano . Open source clustering software. Bioinformatics, Vol. 20 no. 9 2004, pages 1453–1454
32. Wang G., T. Yu and W. Zhang. WordSpy: identifying transcription factor binding motifs by building a dictionary and learning a grammar. Nucleic Acids Research, 2005, Vol. 33, W412–W416
33. The *C. elegans* Sequencing Consortium. "Genome sequence of the nematode *C. elegans*: a platform for investigating biology". Science 282 (5396): 2012–2018,
34. Durbin R., Eddy S., Krogh A., and Mitchison G., Biological Sequence Analysis, Probabilistic models of proteins and nucleic acids, 82005th ed. Cambridge: Cambridge University Press, 1998.
35. Rabiner, L.R. and B. H. Juang, "An Introduction to Hidden Markov Models," IEEE ASSP Magazine, pp. 4-16, January 1986.
36. Tan, Pang-ning, Michael Steinbach, and Vipin Kumar, Introduction to Data Mining. Boston: Pearson Education, Inc., 2006.
37. Genome BioInformatics Research Laboratory. (2005, Oct.) Resources & Datasets. [Online]. http://genome.imim.es/databases/genomics96/index.html
38. Fickett J.W. and C.-S. Tung, "Assessment of protein coding measures.," Nucleic Acids Res. 20:6441–6450, pp. 6441-6450, 1992.
39. Snyder E.E.and Stormo G.D., "Identification of protein coding regions in genomic DNA.," J. Mol. Biol. 248:1-18, pp. 1-18, 1995.
40. Fickett J.W., "The gene identification problem: An overview for developers," Computers Chem. Vol 20, No. 1, pp. 103-118, 1996. [Online]. http://www.nslij-genetics.org/gene/1996.html
41. Rogic S. HMR195 dataset. [Online]. http://www.cs.ubc.ca/~rogic/evaluation/dataset.html
42. WormBase web site, http://www.wormbase.org, release WS200, date 20 Mar 2009. [Online]. http://ws200.wormbase.org/
43. Winters-Hilt S. Using a meta-HMM for Alternative-splice Gene Structure Identification. Paper in Preparation.
44. Du Preez J.A. and D.M. Weber, "High-order hidden Markov modelling," in Communications and Signal Processing, 1998. COMSIG '98. Proceedings of the 1998 South African Symposium on, University of Cape Town, Rondebosch, 7-8 Sept. 1998, pp. 197-202.
45. Reese M.G., Frank H. Eeckman, David Kulp, and David Haussler, "Improved splice site detection in Genie," RECOMB '97: Proceedings of the first annual international conference on Computational molecular biology, pp. 232-240, January, 1997.

Fig. 1

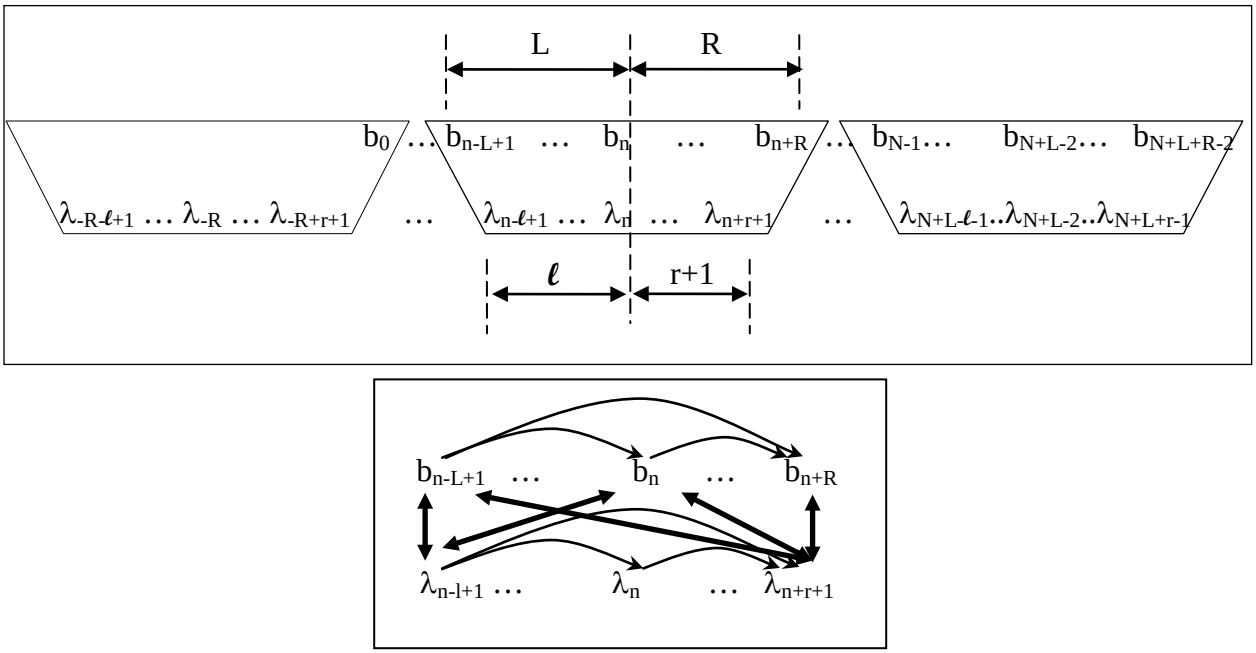

L
R
b0 … bn-L+1 … bn … bn+R … bN-1… bN+L-2… bN+L+R-2
λ-R-ℓ+1 … λ-R … λ-R+r+1 … λn-ℓ+1 … λn … λn+r+1 … λN+L-ℓ-1..λN+L-2..λN+L+r-1
ℓ
r+1
bn-L+1 … bn … bn+R
λn-l+1 … λn … λn+r+1

Fig. 2

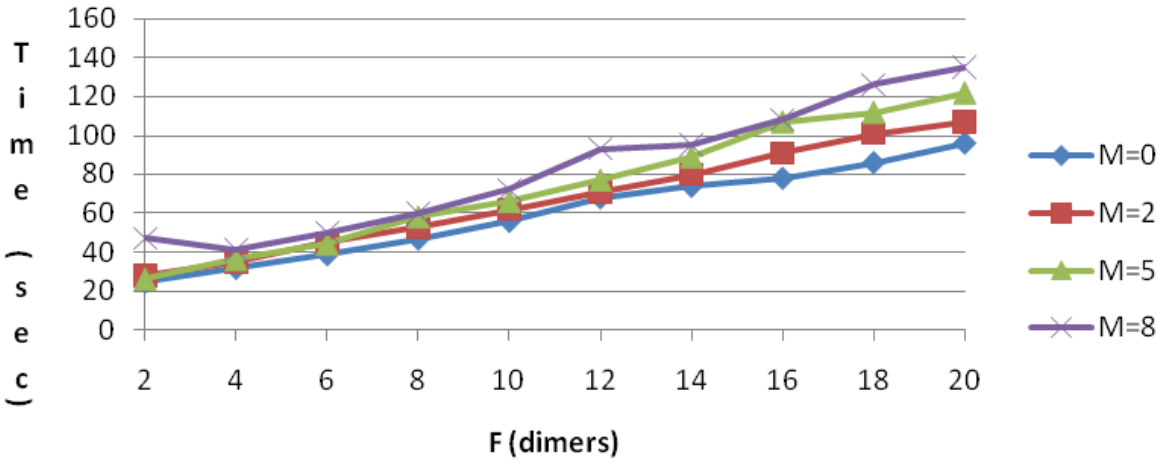

Time (sec)
160
140
120
100
80
60
40
20
0
2
4
6
8
10
12
14
16
18
20
F (dimers)
M=0
M=2
M=5
M=8

Fig. 3

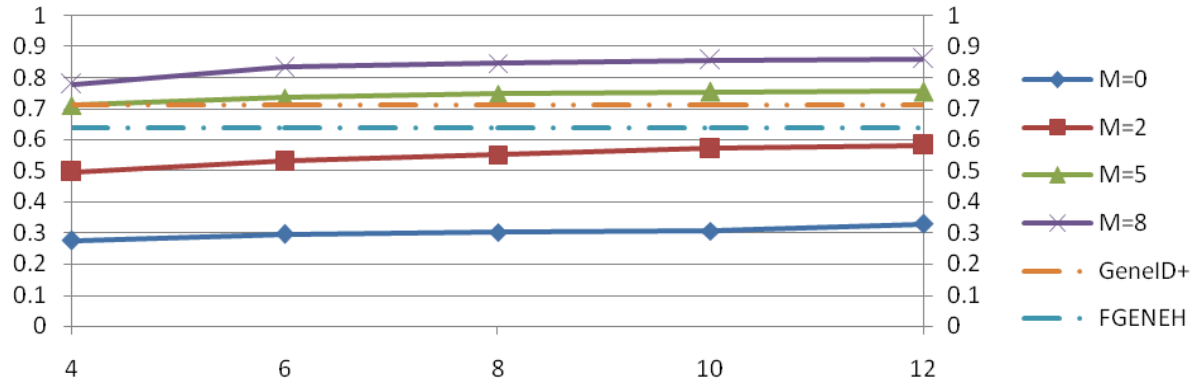


Fig. 4

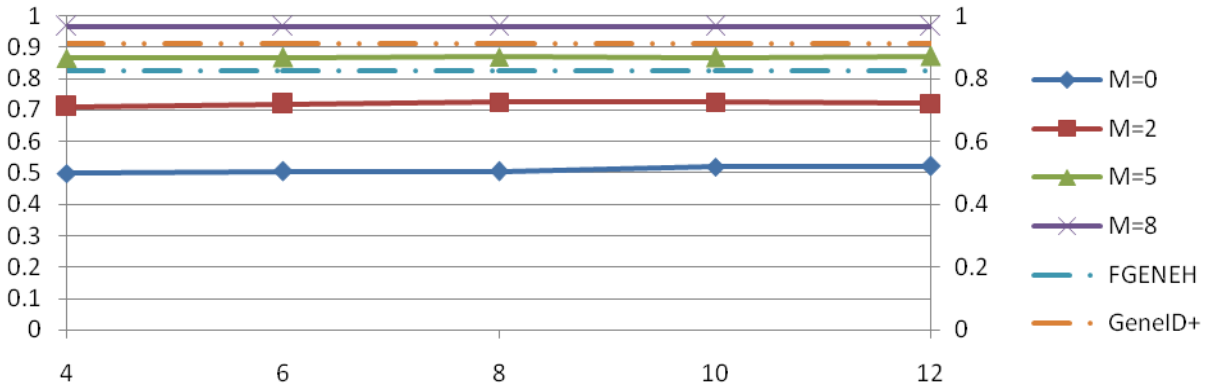


Fig. 5

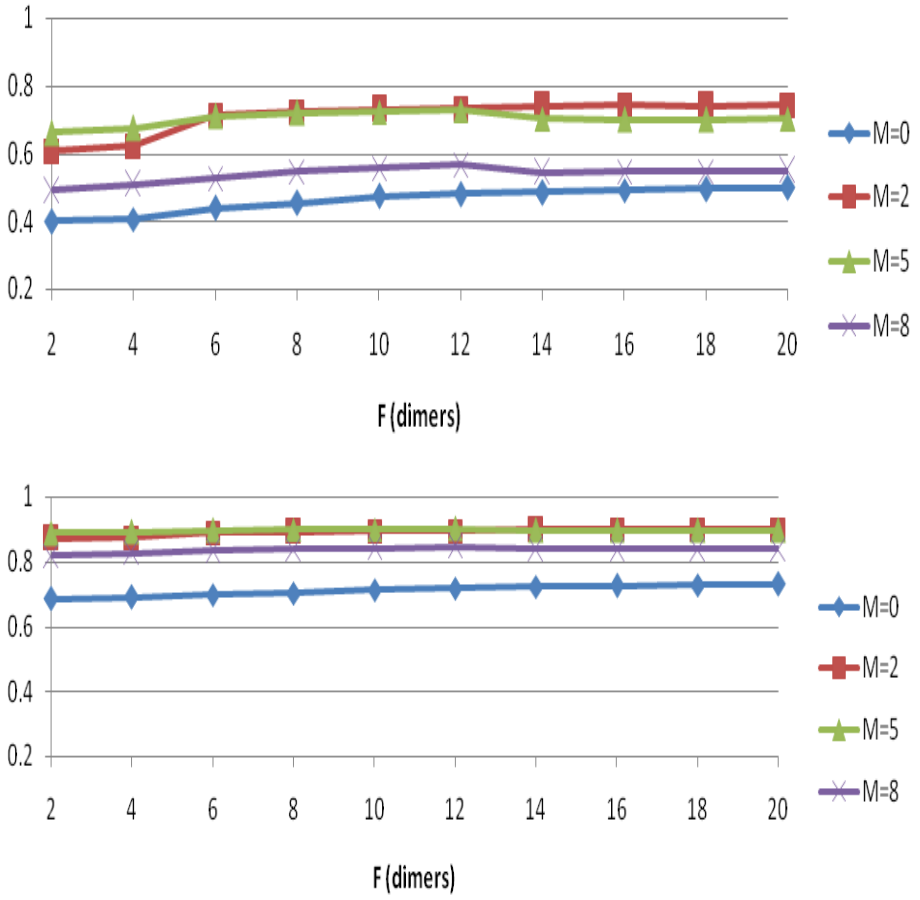
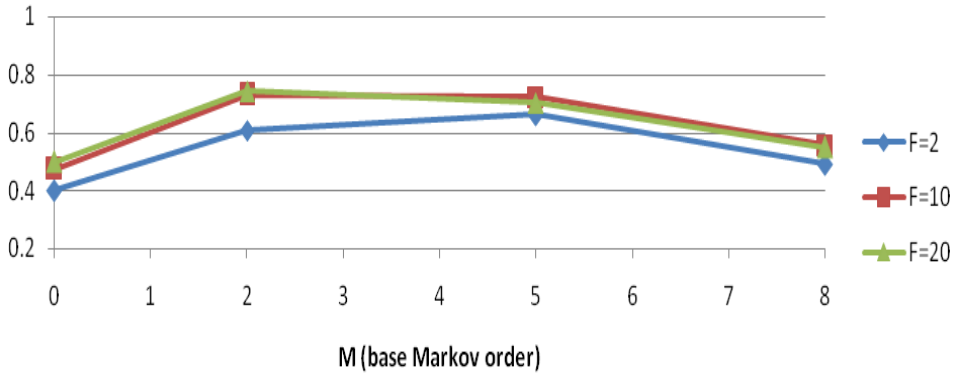
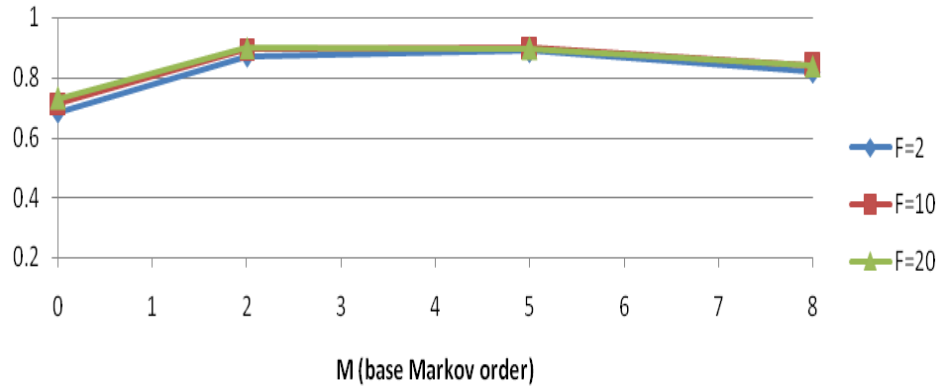

1
0.8
0.6
0.4
0.2
2
4
6
8
10
12
14
16
18
20
F (dimers)
M=0
M=2
M=5
M=8
1
0.8
0.6
0.4
0.2
2
4
6
8
10
12
14
16
18
20
F (dimers)
M=0
M=2
M=5
M=8

Fig. 6


1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
M (base Markov order)
F=2
F=10
F=20


1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
M (base Markov order)
F=2
F=10
F=20