

Using Multiple Alignments to Improve Gene Prediction

Samuel S. Gross^{1,2} and Michael R. Brent¹

¹ Department of Computer Science and Engineering,
Washington University, St. Louis, MO 63130, USA
brent@cse.wustl.edu

² Current address: Computer Science Department,
Stanford University, Stanford, CA 94305, USA
ssgross@cs.stanford.edu

Abstract. The multiple species de novo gene prediction problem can be stated as follows: given an alignment of genomic sequences from two or more organisms, predict the location and structure of all protein-coding genes in one or more of the sequences. Here, we present a new system, N-SCAN (a.k.a. TWINSCAN 3.0), for addressing this problem. N-SCAN has the ability to model dependencies between the aligned sequences, context-dependent substitution rates, and insertions and deletions in the sequences. An implementation of N-SCAN was created and used to generate predictions for the entire human genome. An analysis of the predictions reveals that N-SCAN's predictive accuracy in human exceeds that of all previously published whole-genome de novo gene predictors. In addition, predictions were generated for the genome of the fruit fly *Drosophila melanogaster* to demonstrate the applicability of N-SCAN to invertebrate gene prediction.

1 Introduction

Two recent developments have increased interest in de novo gene prediction. First, the availability of assemblies of several non-human vertebrate genomes has created the possibility for further significant improvements in human gene prediction through the use of comparative genomics techniques. Second, traditional experimental methods for identifying genes based on 5' EST sampling and cDNA clone sequencing are now reaching the point of diminishing returns far short of the full gene set [1]. As a result, efforts to identify new genes by RT-PCR from predicted gene structures are taking on greater importance.

A major advantage of de novo gene predictors is that they do not require cDNA or EST evidence or similarity to known transcripts when making predictions. This allows them to predict novel genes not clearly homologous to any previously known gene, as well as genes that are expressed at very low levels or in only a few specific tissue types, which are unlikely to be found by random sequencing of cDNA libraries. De novo gene predictors are therefore well-suited to

the task of identifying new targets for RT-PCR experiments aimed at expanding the set of known genes.

One of the first *de novo* systems to perform well on typical genomic sequences containing multiple genes in both orientations was GENSCAN [3]. GENSCAN uses a generalized hidden Markov model (GHMM) to predict genes in a given target sequence, using only that sequence as input. GENSCAN remained one of the most accurate and widely used systems prior to the advent of dual-genome *de novo* gene predictors. The initial sequencing of the mouse genome made it possible for the first time to incorporate whole-genome comparison into human gene prediction [2]. This led to the creation of a new generation of gene predictors, such as SLAM [4], SGP2 [5], and TWINSKAN [6, 7, 8], which were able to improve on the performance of GENSCAN by using patterns of conservation between the human and mouse genomes to help discriminate between coding and noncoding regions. These programs are the best-performing *de novo* gene predictors for mammalian genomes currently available.

Recently, there has been an effort to create systems capable of using information from several aligned genomes to further increase predictive accuracy beyond what is possible with two-genome alignments. Programs such as EXONIPHY [9], SHADOWER [10], and the EHMMs of Pedersen and Hein [11] fall into this category. While many important advances have been made in this area, no system of this type has yet managed to robustly outperform two-sequence systems on a genomic scale.

The gene prediction system presented here, N-SCAN (or TWINSKAN 3.0), extends the TWINSKAN model to allow for an arbitrary number of informant sequences as well as richer models of sequence evolution. N-SCAN is descended from TWINSKAN 2.0, which is in turn descended from the GENSCAN GHMM framework. However, instead of emitting a single DNA sequence like GENSCAN or a target DNA sequence and a conservation sequence like TWINSKAN, each state in the N-SCAN GHMM emits one or more columns of a multiple alignment. N-SCAN uses output distributions for the target sequence that are similar to those used by TWINSKAN 2.0 and GENSCAN. It augments these with Bayesian networks which capture the evolutionary relationships between organisms in the multiple alignment. The state diagram is also extended to allow for explicit modeling of 5' UTR structure as well as other conserved noncoding sequence.

2 Methods

2.1 Overview

Whereas TWINSKAN's GHMM outputs a target genomic sequence and a conservation sequence, N-SCAN's GHMM outputs a multiple alignment $\{\mathbf{T}, \mathbf{I}^1, \dots, \mathbf{I}^N\}$ of the target sequence, \mathbf{T} , and the N informant sequences, \mathbf{I}^1 through \mathbf{I}^N . The target sequence consists of the four DNA bases, while the informant sequences can also contain the character “_”, representing gaps in the alignment

and “.”, representing positions in the target sequence to which the informant sequence does not align. The states in the N-SCAN GHMM correspond to functional categories in the target sequence only. Therefore, N-SCAN annotates only one sequence at a time. If annotations are desired for more than one of the sequences in the alignment, the system can be run multiple times with different sequences designated as the target.

One component of the model defines, for each GHMM state, the probability

$$P(T_i|T_{i-1}, \dots, T_{i-o}) \quad (1)$$

of outputting a particular base in the target genome at the current position in the sequence, given the previous o bases. Here T_1, \dots, T_L is the full target sequence \mathbf{T} and o is the model order. This probability is implicitly dependent on the GHMM state at base i . States represent sequence features, such as start and stop codons, splice sites, and coding sequence. N-SCAN uses these target genome models in combination with a set of Bayesian networks to define, for each state, the probability

$$P(T_i, I_i^1, \dots, I_i^N | T_{i-1}, I_{i-1}^1, \dots, I_{i-1}^N, \dots, T_{i-o}, I_{i-o}^1, \dots, I_{i-o}^N) \quad (2)$$

of outputting a column in the alignment given the previous o columns. This is accomplished by multiplying the probability from the target genome model by

$$P(I_i^1, \dots, I_i^N | T_i, T_{i-1}, I_{i-1}^1, \dots, I_{i-1}^N, \dots, T_{i-o}, I_{i-o}^1, \dots, I_{i-o}^N) \quad (3)$$

This quantity can be computed from the Bayesian network associated with the state.

We assume that the probability of outputting a base in the target sequence is independent of the values of the previous o positions in all of the informants, given the values of the previous o positions in the target. That is,

$$P(T_i|T_{i-1}, \dots, T_{i-o}) = P(T_i|T_{i-1}, I_{i-1}^1, \dots, I_{i-1}^N, \dots, T_{i-o}, I_{i-o}^1, \dots, I_{i-o}^N) \quad (4)$$

Given (4), we can multiply (1) by (3) to obtain (2).

2.2 Phylogenetic Bayesian Networks

The Bayesian network representation used in N-SCAN is similar to the phylogenetic models described in [12], with a few important differences. First, the N-SCAN model uses a six-character alphabet consisting of the four DNA bases plus characters representing gaps and unaligned positions. In addition, the substitutions between nodes in the model need not take place via a continuous time Markov process. Finally, the two models use slightly different underlying graphs. For now, we will only discuss Bayesian networks that define a distribution involving single columns of a multiple alignment; context-dependence will be introduced later.

Consider a phylogenetic tree such as the one shown in Fig. 1, left. Leaf nodes represent present-day species, while non-leaf nodes represent ancestral species

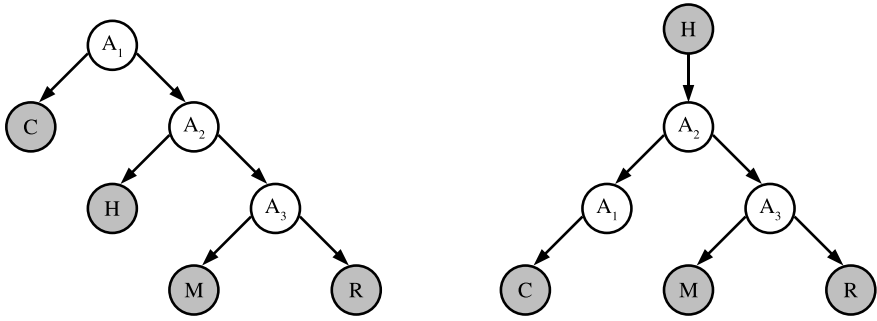


Fig. 1. A phylogenetic tree relating chicken (C), human (H), mouse (M), and rat (R). The graph can also be interpreted as a Bayesian network (left). The result of transforming the Bayesian network (right)

which no longer exist. The same graph can also be interpreted as a Bayesian network describing a probability distribution over columns in a multiple alignment. In that case, the nodes represent random variables corresponding to characters at specific rows in a multiple alignment column and the edges encode conditional independence relations among the variables. The independencies represented by the phylogenetic tree are quite natural – once we know the value of the ancestral base at a particular site in the alignment column, the probabilities of the bases in one descendant lineage are independent of the bases in other descendant lineages. These independence relations allow us to factor the joint distribution as follows:

$$P(H, C, M, R, A_1, A_2, A_3) = P(A_1) \cdot P(C|A_1) \cdot P(A_2|A_1) \cdot P(H|A_2) \cdot P(A_3|A_2) \cdot P(M|A_3) \cdot P(R|A_3)$$

By taking advantage of the conditional independence relations present in the seven-variable joint distribution, we can express it as a product of six local conditional probability distributions (CPDs) that have two variables each and a marginal distribution on one variable. In general, factoring according to the independencies represented by a phylogenetic tree leads to an exponential reduction in the number of parameters required to specify the joint distribution. Of course, a real multiple alignment will only consist of sequences from currently existing species. Therefore, we treat the ancestral variables as missing data for the purposes of training and inference (see below). Rather than using a Bayesian network with the same structure as the phylogenetic tree, however, we apply a transformation to the phylogenetic tree graph to create the Bayesian network structure used in N-SCAN.

To transform the graph, we reverse the direction of all the edges along the path from the root of the graph to the target node. This results in a new Bayesian network with a tree structure rooted at the target node (see Fig. 1, right). The new Bayesian network encodes the same conditional independence relations as the original, but it suggests a new factorization of the joint distribution. For the example network shown in Fig. 1, this factorization is:

$$P(H, C, M, R, A_1, A_2, A_3) = P(H) \cdot P(A_2|H) \cdot P(A_1|A_2) \cdot P(A_3|A_2) \cdot P(C|A_1) \cdot P(M|A_3) \cdot P(R|A_3)$$

In this factorization, the local distribution at the node corresponding to the target sequence ($P(H)$ in the example) is not conditioned on any of the other variables. This allows us to directly use existing single-sequence gene models to account for the effect of the target sequence on the probability of alignment columns. Previous attempts to integrate phylogenetic trees and HMMs have used a prior distribution on the unobserved common ancestor sequence.

One final alteration to the Bayesian network is made after the transformation described above. Any ancestral node with just one child is removed from the network along with its impinging edges. For each removed node, a new edge is added from the removed node's parent to its child. In the example, we remove A_1 and add an edge from A_2 to C . Again, it is not difficult to show that this transformation does not affect the expressive power of the network. We can write the local CPD at the removed node's child as a sum over the all possible values of the removed node. In the example,

$$P(C|A_2) = \sum_{A_1} P(C|A_1)P(A_1|A_2)$$

In effect, we have implicitly summed out some of the unobserved variables in the distribution. In general, we are only interested in computing the probability of an assignment to the observed variables. When making such a computation, we explicitly sum out all the unobserved variables in the distribution. The transformation described above makes this computation more efficient by reducing the number of explicit summations required.

2.3 Context-Dependent Models

Following [12], we can extend the models presented above to incorporate context dependence by redefining the meaning of the variables in the network. For a model of order o , we interpret the random variables in the network to represent the value of $o+1$ adjacent positions in a row in the alignment. The entire network then defines a joint distribution over sets of $o+1$ adjacent columns, which can be used to determine the probability of a single column given the previous o columns.

Inference in the network can be accomplished using a modified version of Felsenstein's algorithm [13]. First, consider the problem of calculating the probability of an assignment to all the informant nodes in the network, given the value of the target node. For a given assignment, we define $L_u(a)$ to be the joint probability of all the observed variables that descend from node u , given that node u has value a . If $C(u)$ is the set of children of u and $V(u)$ is the set of possible values of u , we can calculate $L_u(a)$ according to the following recursive formula:

$$L_u(a) = \begin{cases} M(u, a) & \text{if } u \text{ is a leaf} \\ \prod_{c \in \mathcal{C}(u)} \left(\sum_{b \in V(c)} Pr(c = b | u = a) L_c(b) \right) & \text{otherwise} \end{cases}$$

Here, M is called the match function, and is defined as follows:

$$M(u, a) = \begin{cases} 1 & \text{if node } u \text{ has value } a \\ 0 & \text{otherwise} \end{cases}$$

If T is the target node, and t is its observed assignment, then $L_T(t)$ is the probability of the informant assignments given the target. To calculate all the L_u 's for each node in the network, we can visit the nodes in postorder and calculate all the L_u 's for a particular node at once.

We can use essentially the same algorithm for a conditional probability query. We define the partial match function, M' , for a model of order o as follows:

$$M'(u, a) = \begin{cases} 1 & \text{if the first } o \text{ characters of the value of node } u \\ & \text{match the first } o \text{ characters of } a \\ 0 & \text{otherwise} \end{cases}$$

We define the quantity $L'_u(a)$ exactly as we did $L_u(a)$, except we substitute M' for M in the recursive definition. $L'_T(t)$ is then the probability of the first o characters of the informant assignments. Once we know the values of $L_T(t)$ and $L'_T(t)$, expression (3) is just

$$\frac{L_T(t)}{L'_T(t)}$$

Each call to the inference algorithm visits each node in the network exactly once, and requires $O(6^{2(o+1)})$ operations per internal node. Thus, the overall time complexity of inference is $O(N \cdot 6^{2(o+1)})$. Adding additional informants only results in a linear increase in the complexity of inference, but we pay an exponential cost for increasing the model order.

2.4 Training

The Bayesian networks for all of N-SCAN's GHMM states share a single topology determined by the phylogenetic tree relating the target and the informant genomes, which we assume to be known. However, the local CPDs for each node in a particular network will depend on the GHMM state with which the network is associated. The CPDs are not known in advance, and must be estimated from training data.

Suppose we had a multiple alignment of all the genomes represented in the phylogenetic tree, with each column labeled to indicate which GHMM state produced it. For a particular Bayesian network of order o , we could treat each set of $o + 1$ adjacent columns ending with a column labeled by the GHMM state associated with the network as an instantiation of the network variables. Once

we extract a list of all the instantiations that occur in the multiple alignment, along with the number of times each instantiation occurs, it is a simple matter to produce a maximum likelihood estimate for all the CPDs in the network.

Since the GHMM states correspond to gene features, we can construct a labeled multiple alignment by combining the output of a whole-genome multiple aligner with a set of annotations of known genes. However, the alignment will contain only the genomes that correspond to the root and leaves of the Bayesian network graph. The ancestral genomes are no longer available for sequencing and so must be treated as missing data.

We can still estimate the CPDs despite the missing data by using the EM algorithm. For each network, we begin with an initial guess for the CPDs. We then calculate, for each CPD, the expected number of times each possible assignment to its variables occurs in the multiple alignment. This can be done efficiently using a variation of the inside-outside algorithm essentially the same as the one presented in [12]. Next, the initial guess is replaced with a maximum likelihood estimate of the CPDs based on the expected occurrences. This process is repeated until the maximum likelihood estimate converges. At convergence, the maximum likelihood estimate is guaranteed to be a stationary point of the likelihood function of the multiple alignment.

2.5 CPD Parameterizations

We have not yet described a method for obtaining a maximum likelihood estimate of the CPDs from a set of observations (or expected observations). If no restrictions are placed on the form taken by the CPDs, there exist simple closed-form expressions for the value of each entry in each CPD. A Bayesian network with completely general CPDs can represent any joint distribution in which the conditional independence relations it encodes hold. However, a complete description of such a network requires a relatively large number of parameters. Each N-SCAN Bayesian network of order o has $(2N - 1)(6^{o+1})(6^{o+1} - 1)$ free parameters if its CPDs are unrestricted. If the amount of training data (i.e., columns in the multiple alignment with the appropriate label) available is small, this may be too many parameters to fit accurately.

It is possible to reduce the number of parameters to fit by specifying the CPDs using fewer than $(6^{o+1})(6^{o+1} - 1)$ parameters each. Only a subset of all possible CPDs will be expressible by any given non-general parameterization, but we hope the real CPDs, or ones close to them, will be expressible by the parameterization we choose. Depending on the parameterization chosen, we may be able to derive analytical expressions with which to estimate the values of the parameters. Otherwise, we can use numerical optimization techniques to obtain an estimate.

In the experiments below, we use a parameterization with a form similar to the general reversible rate matrices used in traditional phylogenetic models. The zero-order version of this parameterization, which we call a *partially reversible* model, is shown below. A cell (i, j) in the matrix represents $P(j|i)$, the probability of a particular child node having value j from the alphabet $\{A, C, G, T, -, .\}$, given that its parent has value i .

$$\begin{pmatrix} - & a\pi_C & b\pi_G & c\pi_T & g & h \\ a\pi_A & - & d\pi_G & e\pi_T & g & h \\ b\pi_A & d\pi_C & - & f\pi_T & g & h \\ c\pi_A & e\pi_C & f\pi_G & - & g & h \\ i\pi_A & i\pi_C & i\pi_G & i\pi_T & - & j \\ k\pi_A & k\pi_C & k\pi_G & k\pi_T & l & - \end{pmatrix}$$

Here, the π_i 's are the background frequency of the bases; they are estimated directly from the multiple alignment and are not considered to be free parameters. The model has 12 free parameters, as opposed to 30 in a general parameterization. Note that the 4x4 upper-left submatrix is identical to the general reversible rate matrix used in continuous time Markov process models of sequence evolution [14]. The probability of a deletion is the same for each base, as is the probability of a base becoming unaligned. The probability of a base being inserted or becoming realigned is proportional to the background frequency of the base.

To generalize the partially reversible parameterization to higher orders, we make use of the concept of a gap pattern. We define the gap pattern of an $(o + 1)$ -mer to be the string that results from replacing all the bases in the $(o + 1)$ -mer with the character "X". For example, the trimers "GA_", "GC_", and "AT_" all have the gap pattern "XX_". For substitution probabilities involving an $(o + 1)$ -mer that contain gaps or unaligned characters, the partially reversible model considers only the gap pattern of the $(o + 1)$ -mer. Let \mathcal{D} be the set of all possible $(o + 1)$ -mers that contain only the four DNA bases, and \mathcal{G} be the set of all possible gap patterns of length $o + 1$ that contain at least one gap or unaligned character. The substitution probabilities $P(j|i)$ have the following properties:

1. If $j \in \mathcal{D}$ and $i \in \mathcal{D}$, then $P(j|i)\pi_i = P(i|j)\pi_j$.
2. If $j \in \mathcal{D}$ and $i \in \mathcal{G}$, then $P(j|i) = \alpha_i\pi_j$.
3. If $j \in \mathcal{G}$ and $i \in \mathcal{D}$, then $P(j|i) = \beta_j$.

It can be shown that a sequence evolving according to a substitution process that has these three properties will have constant expected values for the relative frequencies of the $(o + 1)$ -mers in \mathcal{D} . The first-order partially reversible model, which can be described by a 36x36 matrix, has 170 free parameters, far fewer than the 1260 in the general first-order model.

Partially reversible models are able to capture significantly more information about patterns of selection than the conservation sequence approach used in TWINSKAN 2.0, which considers only patterns of matches, mismatches, and unaligned positions. For example, a first-order partially reversible model can model insertions and deletions separately from base substitutions, and can take into account the difference between the rates of transitions and transversions as well as the increased rate of mutation of CpG dinucleotides [15]. Furthermore, unlike TWINSKAN 2.0, N-SCAN uses a separate conservation model for each

codon position in coding sequence, allowing it to model differences in substitution rates between the three positions.

2.6 Conservation Score Coefficient

Like TWINSKAN, N-SCAN uses log-likelihood scores rather than probabilities internally. The score of a particular column i in the multiple alignment given a state S can be written as

$$\log \left(\frac{T_S(i)}{T_{Null}(i)} \right) + k \cdot \log \left(\frac{C_S(i)}{C_{Null}(i)} \right)$$

Here, T_S and T_{Null} are the target sequence probabilities of the form shown in expression (1) for state S and the null model, respectively. Likewise, C_S and C_{Null} are the conservation model probabilities, as in expression (3). k is an arbitrary constant called the conservation score coefficient which can be used to increase or decrease the impact of the informant sequences on N-SCAN's predictions. Empirical results show that a value of k between 0.3 and 0.6 leads to the best predictive performance. This may be due to the potential of conserved noncoding regions to contribute to a large number of false positive predictions (see below).

2.7 State Diagram

Figure 3 shows the N-SCAN state diagram. The 5' UTR and CNS states allow N-SCAN to avoid false positives that would occur if these sequence features were not modeled explicitly. Without these states, conserved noncoding regions would tend to be annotated as coding exons due to their high conservation scores. Instead, N-SCAN tends to annotate conserved regions with a low coding score as CNS. Furthermore, the 5' UTR states allow N-SCAN to predict exon/intron structure in 5' UTRs. Simultaneous 5' UTR and coding region prediction by N-SCAN will be discussed in more detail in a forthcoming paper devoted to the subject [16].

Since we lacked a reliable set of annotations of conserved noncoding regions, we used the null target sequence model for the CNS state, which effectively assigns the neutral score of zero to all target sequences under the CNS model. Thus, the score of a putative CNS region is determined entirely by the CNS conservation model, which was estimated from 5' UTRs. While this resulted in an acceptable model for some types of CNS, more highly-conserved CNS was probably not modeled accurately using this method.

2.8 Experimental Design

The human gene prediction experiments presented below were performed on the May 2004 build of the human genome (hg17), and the January 2003 build of the

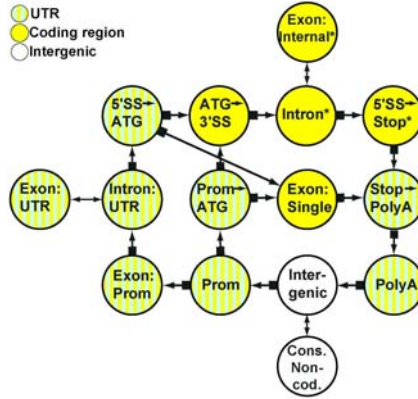


Fig. 2. The N-SCAN state diagram. Intron and exon states with asterisks represent six states each, which are used tracking reading frame and partial stop codons. Only forward strand states are shown; on the reverse strand all non-intergenic states are duplicated, and initial (ATG \rightarrow 3' SS), not terminal (5' SS \rightarrow Stop), states track phase and partial stop codons

Drosophila melanogaster genome. Both were obtained from the UCSC genome browser [17]. Each group of experiments used a set of annotations consisting of known genes, which was constructed as follows. The annotation set initially contained the mappings of RefSeqs to the genome in question provided by the UCSC genome browser. This set was then filtered to exclude annotations believed likely to have errors. All genes with non-standard start or stop codons, in-frame stop codons, total coding region length not a multiple of three, non-standard donor sites lacking a GT, GC, or AT consensus, or non-standard acceptor sites lacking an AG or AC consensus were discarded. After filtering, the human set contained 16,259 genes and 20,837 transcripts, while the *D. melanogaster* set contained 13,091 genes and 18,591 transcripts.

For the human experiments, N-SCAN used an eight-way whole-genome alignment of human (hg17), blowfish (fr1), chicken (galGal2), chimp (panTro1), dog (canFam1), mouse (mm5), rat (rn3), and zebrafish (danRer1) created by MULTIZ [18]. A four-way MULTIZ alignment of *Drosophila melanogaster*, *Drosophila yakuba*, *Drosophila pseudoobscura*, and *Anopheles gambiae* was used for the *D. melanogaster* experiments. The alignments were downloaded from the UCSC genome browser. For the human experiments, we used only the rows of the alignment corresponding to human, chicken, mouse, and rat, and discarded the other four rows. Columns in either alignment with gaps in the target sequence were also discarded.

All predictions made by N-SCAN were four-fold cross validated. The first-order partially reversible parameterization was used for all of N-SCAN's Bayesian network CPDs, and N-SCAN's conservation score coefficient was set to 0.4.

3 Results

3.1 Human Gene Prediction Performance Comparison

To evaluate the predictive performance of N-SCAN, we generated predictions for every chromosome in the hg17 human genome assembly. We then compared the N-SCAN predictions, as well as the predictions of several other de novo gene predictors, to our test set of known genes. The gene prediction systems involved in this experiment included one single-genome predictor (GENSCAN), two dual-genome predictors (SGP2 and TWINSKAN 2.0), and two multiple-genome predictors (EXONIPHY and N-SCAN). SGP2 and TWINSKAN 2.0 made use of human-mouse alignments, while EXONIPHY and N-SCAN used multiple alignments of human, chicken, mouse, and rat. The GENSCAN predictions used in this experiment were downloaded from the UCSC genome browser. The SGP2 predictions were downloaded from the SGP2 web site [19]. The EXONIPHY predictions were obtained from one of EXONIPHY's creators (A.C. Siepel, personal communication). EXONIPHY does not link exons into gene structures, so its performance at the gene level was not evaluated.

We evaluated both sensitivity and specificity at the gene, transcript, exon, and nucleotide levels. Since none of the gene predictors involved in the experiment had the ability to predict alternative transcripts, a gene prediction was counted as correct at the gene level if it exactly matched any of the transcripts in the test set. The results of the experiment are shown in Table 1. For each performance metric, the result from the best-performing predictor is shown in bold. Note that the specificity numbers are underestimates, since all predicted genes not in the test set were counted as incorrect. N-SCAN achieved substantially better performance on both the gene and exon levels than the other four predictors involved in the experiment. On the nucleotide level, N-SCAN had the highest sensitivity, but a lower specificity than EXONIPHY.

We also evaluated the ability of the systems to predict long introns, a feat notoriously difficult for de novo gene predictors. The results in Table 2 show that N-SCAN has the greatest sensitivity for each length range we tested. Furthermore, N-SCAN's performance drops off much more slowly with length than

Table 1. Whole-genome gene prediction performance in human

	GENSCAN	TWINSKAN 2.0	N-SCAN	SGP2	EXONIPHY
Gene Sn	0.10	0.25	0.35	0.16	-
Gene Sp	0.04	0.15	0.21	0.08	-
Transcript Sn	0.08	0.21	0.29	0.14	-
Transcript Sp	0.04	0.15	0.21	0.08	-
Exon Sn	0.69	0.71	0.84	0.73	0.57
Exon Sp	0.33	0.61	0.63	0.52	0.50
Nucleotide Sn	0.86	0.84	0.90	0.86	0.76
Nucleotide Sp	0.40	0.64	0.65	0.62	0.68

Table 2. Intron sensitivity by length

Length (Kb)	Count	GENSCAN	TWINSKAN 2.0	N-SCAN	SGP2
0 - 10	157757	0.68	0.77	0.86	0.74
10 - 20	9519	0.50	0.46	0.77	0.69
20 - 30	3317	0.41	0.22	0.71	0.68
30 - 40	1742	0.30	0.08	0.64	0.60
40 - 50	992	0.28	0.02	0.64	0
50 - 60	652	0.20	0.01	0.53	0
60 - 70	447	0.16	0	0.52	0
70 - 80	314	0.12	0	0.50	0
80 - 90	268	0.10	0	0.39	0
90 - 100	211	0.11	0	0.48	0

Table 3. Performance with different combinations of informants. The human informants are chicken (C), mouse (M), and rat (R). The *D. melanogaster* informants are *A. gambiae* (G), *D. Pseudoobscura* (P), and *D. Yakuba* (Y)

	Human				<i>D. melanogaster</i>			
	C	M	R	C, M, R	G	P	Y	G, P, Y
Gene Sn	0.21	0.34	0.32	0.34	0.39	0.53	0.49	0.55
Gene Sp	0.16	0.23	0.23	0.22	0.39	0.52	0.48	0.53
Transcript Sn	0.18	0.29	0.27	0.29	0.32	0.43	0.40	0.45
Transcript Sp	0.16	0.23	0.23	0.22	0.39	0.52	0.48	0.53
Exon Sn	0.75	0.81	0.80	0.82	0.67	0.74	0.73	0.77
Exon Sp	0.57	0.61	0.62	0.61	0.67	0.76	0.73	0.75
Nucleotide Sn	0.87	0.87	0.88	0.88	0.92	0.93	0.92	0.93
Nucleotide Sp	0.61	0.63	0.64	0.64	0.92	0.94	0.94	0.94

that of the other gene predictors. In fact, N-SCAN is able to correctly predict approximately half of the introns in the test set with lengths between 50Kb and 100Kb.

3.2 Informant Effectiveness

To test the effect of multiple informants on N-SCAN's predictive accuracy, we generated four sets of predictions each for human and *D. melanogaster*. The first three sets for each organism use a single informant, while the final set uses all three informants simultaneously. Predictions were generated for human chromosomes 1, 15, 19, 20, 21, and 22, and for the entire *D. melanogaster* genome. The results of this experiment are shown in Table 3. In *D. melanogaster*, N-SCAN achieved a small but significant boost in performance by using all three informants together. However, in human, using the three informants at once appears to be no better than using mouse alone.

4 Discussion

We have presented a system, N-SCAN, for de novo gene prediction that builds on an existing system by incorporating several new features, such as richer substitution models, states for 5' UTR structure prediction, a conserved noncoding sequence state, and the ability to use information from multiple informant sequences. N-SCAN achieved significantly better performance than several other de novo gene predictors in a test of whole-genome gene prediction in human. In addition, N-SCAN was successfully applied to gene prediction in *D. melanogaster* without the need for any special modifications.

N-SCAN incorporates information from multiple informant sequences in a novel way which we believe has several potential advantages. First, N-SCAN builds on existing single-sequence models of a target genome. These single-sequence models can be quite sophisticated. For example, the donor splice site model used in GENSCAN and TWINSCAN 2.0 is able to take into account the effect of non-adjacent positions in the splice site signal through the use of a maximal dependence decomposition model [3]. In addition, single-sequence models of a given order generally require fewer parameters than multiple-sequence models of the same order. Therefore, it is possible to use high-order single-sequence models in combination with conservation models of a lower order while maintaining a good fit. In the experiments presented above, some of the N-SCAN target genome models had orders as high as five, while the conservation models were all of order one. Furthermore, because the target sequence is observed, it is possible to obtain a globally optimal estimate of the distributions in the target genome models. The EM estimates for the conservation models are only guaranteed to be locally optimal, and could in principle be far from a global optimum.

Also important is N-SCAN's treatment of gaps and unaligned characters. Instead of treating these characters as missing data, or modeling gap patterns using additional states in the GHMM [9], N-SCAN deals with them directly in its conservation models. This allows the very significant information they contribute to be taken into account in a natural and efficient way. The price for this ability is that the continuous time Markov process model of substitution must be abandoned. Continuous time Markov processes are a good model for base mutations between aligned positions in DNA sequences, but do not accurately model the nonlinear process of positions becoming unaligned over time. For the sake of illustration, consider an ancestor sequence and a descendant sequence that differ by only a single point mutation. It is not possible for the sequences to have unalignable bases. The alignment will have a gap if the single mutation is an insertion or deletion, but the surrounding regions will provide enough information to align the gap with the right base in the other species. Thus, the instantaneous rates of substitutions leading to unaligned characters are all zero. Yet as divergence increases, a point will be reached where even small changes to the sequence can lead to a whole region becoming unalignable. Therefore, rather than assuming substitutions occur as a result of a continuous time Markov process, N-SCAN uses the more general framework of Bayesian networks for its conservation models. This results in a substantial increase in the required num-

ber of parameters, but with appropriate parameterizations, this number is still manageable for context-dependent models of the type presented here.

Relaxing the assumption that substitutions occur via a continuous time Markov process also allows N-SCAN to accurately model alignment columns in which the aligned positions do not share a single functional state. In such a case, patterns of substitution across different branches of the phylogenetic tree are likely to vary significantly, reflecting different evolutionary constraints. This situation cannot be represented by a continuous time Markov process model, which uses the same substitution rate matrix for each branch in the tree. In practice, positions in an alignment column may have different functions as a result of a function-changing mutation, alignment error, or sequencing error. The latter two causes are of particular concern when the alignment contains highly diverged or draft-quality sequences.

We are currently pursuing a number of approaches for improving N-SCAN. First, the use of higher-order conservation models has the potential to increase N-SCAN's predictive accuracy. Second-order models for coding sequence, for example, could perfectly distinguish between silent and missense mutations. Although both inference and training are far more expensive in second-order models than in first-order models, the use of second-order models for gene prediction in human and *D. melanogaster* appears feasible. Second, better models of conserved noncoding sequence should lead to better performance and perhaps remove the need for a conservation score coefficient. Finally, the role of multiple informants merits further investigation. Although the use of multiple informants in *D. melanogaster* gene prediction improved performance beyond what was achieved with any single informant, the same effect was not observed in human. This may be due to the specific characteristics of the informant sequences that were used for each organism, differences in the properties of the target genomes, or some other factor. Future experiments on a variety of target genomes using different combinations of informants should shed some light on this issue.

Acknowledgments

We thank Mikhail Velikanov for providing the set of filtered RefSeq genes used for training and evaluation. We also thank Adam Siepel for converting the EX-ONIPHY predictions on hg16 to hg17 coordinates. Finally, thanks to all the members of the Brent lab who developed and maintained the TWINSKAN code base, from which N-SCAN was developed. This work was supported by grant HG02278 from the NIH to M.R.B.

References

1. The MGC Project Team. 2004. The status, quality, and expansion of the NIH full-length cDNA project: The Mammalian Gene Collection (MGC). *Genome Res.*, 14:2121-2127.

2. Waterston et al. 2002. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420:520-562.
3. C. Burge and S. Karlin. 1997. Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, 268:78-94.
4. M. Alexandersson, S. Cawley, and L. Pachter. 2003. SLAM: Cross-species gene finding and alignment with a generalized pair hidden Markov model. *Genome Res.*, 13:496-502.
5. G. Parra, P. Agarwal, J.F. Abril, T. Wiehe, J.W. Fickett, and R. Guigo. 2003. Comparative gene prediction in human and mouse. *Genome Res.*, 13:108-117.
6. I. Korf, P. Flicek, D. Duan, and M.R. Brent. 2001. Integrating genomic homology into gene structure prediction. *Bioinformatics*, 17 Suppl. 1:S140-148.
7. P. Flicek, E. Keibler, P. Hu, I. Korf, and M.R. Brent. 2003. Leveraging the mouse genome for gene prediction in human: From whole-genome shotgun reads to a global syntenic map. *Genome Res.*, 13:46-54.
8. A.E. Tenney, R.H. Brown, C. Vaske, J.K. Lodge, T.L. Doering, and M.R. Brent. 2004. Gene prediction and verification in a compact genome with numerous small introns. *Genome Res.*, in press.
9. A.C. Siepel and D. Haussler. 2004. Computational identification of evolutionary conserved exons. In *RECOMB 2004*.
10. J.D. McAuliffe, L. Pachter, and M.I. Jordan. 2003. Multiple-sequence functional annotation and the generalized hidden Markov phylogeny. Technical Report 647, Department of Statistics, University of California, Berkeley.
11. J.S. Pedersen and J. Hein. 2003. Gene finding with a hidden Markov model of genome structure and evolution. *Bioinformatics*, 19:219-227.
12. A. Siepel and D. Haussler. 2004. Phylogenetic estimation of context-dependent substitution rates by maximum likelihood. *Mol. Biol. Evol.*, 21:468-448.
13. Felsenstein, J. 1981. Evolutionary trees from DNA sequences. *J. Mol. Evol.*, 17:368-376.
14. P. Lió and N. Goldman. 1998. Models of molecular evolution and phylogeny. *Genome Res.*, 8:1233-1244.
15. M. Bulmer. 1986. Neighboring base effects on substitution rates in pseudogenes. *Mol. Biol. Evol.*, 3:322-329.
16. R.H. Brown, S.S. Gross, and M.R. Brent. 2005. Begin at the beginning: predicting genes with 5' UTRs. Submitted.
17. W.J. Kent, C.W. Sugnet, T.S. Furey, K.M. Roskin, T.H. Pringle, A.M. Zahler, and D. Haussler. 2003. The human genome browser at UCSC. *Genome Res.*, 12:996-1006.
18. M. Blanchette, W.J. Kent, C. Riemer, L. Elnitski, A.F.A. Smith, K.M. Roskin, R. Baertsch, K. Rosenbloom, H. Clawson, E.D. Green, D. Haussler, and W. Miller. 2004. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res.*, 14:708-715.
19. SGP2 home page. (<http://genome.imim.es/software/sgp2>).