From R. Altman *et al.* (eds.), *Proc. Pacific Symposium on Biocomputing* 5:377–388 (2000)

STOCHASTIC SEGMENT MODELS OF EUKARYOTIC PROMOTER REGIONS

UWE OHLER, GEORG STEMMER, STEFAN HARBECK, HEINRICH NIEMANN
Lehrstuhl für Mustererkennung, Universität Erlangen–Nürnberg
Martensstr.3, D-91058 Erlangen, Germany
eMail: ohler@informatik.uni-erlangen.de

We present a new statistical approach for eukaryotic polymerase II promoter recognition. We apply stochastic segment models in which each state represents a functional part of the promoter. The segments are trained in an unsupervised way. We compare segment models with three and five states with our previous system which modeled the promoters as a whole, i. e. as a single state. Results on the classification of a representative collection of human and *D. melanogaster* promoter and non-promoter sequences show great improvements. The practical importance is demonstrated on the mining of large contiguous sequences.

1 Introduction

As the large sequencing projects, e. g. those of man and *Drosophila*, enter the final stage, we are in urgent need of computer methods to analyze and annotate the large amounts of contiguous genomic sequences. A particularly hard problem is the reliable recognition of transcription start sites (TSS) and/or the promoter regions of genes within genomic DNA. Nothing has essentially changed since Fickett and Hatzigeorgiou stated that this problem is far from being solved [1].

Recently, we presented a content-based system for promoter identification [2] which used no background knowledge about the structural properties of promoter regions (reviewed for example by Kornberg [3] or Nikolov and Burley [4]). We only assumed a window size of 300 bases (250 before and 50 after the TSS) which is the region known to contain most of the transcription factor binding sites involved. This is opposed to the signal-based approaches which look for specific occurrences of transcription elements [5].

We have designed a new hybrid approach which is based on the observation that a eukaryotic promoter can generally be divided into segments: the region upstream from the transcription start site, the core promoter where the main initiation complex binds, and a region downstream from the start site. The core promoter can be further split into the TATA box and the initiator region (Inr), separated by a spacer of approximately 15 bp. We use this broad seg-

mentation of a PolII promoter region to pursue a new approach for promoter recognition based on a stochastic modeling of promoter segments. Our aim is to incorporate as much general structural knowledge as possible without getting as specific as signal-based methods. Previous hybrid approaches [6,7] combined N-mer statistics of several regions upstream of the TSS with a weight matrix for the TATA box, or performed a quadratic discriminant analysis based on feature variables calculated within several windows around the TSS.

In the following, we describe stochastic segment models of *D. melanogaster* and human promoter regions. The model type that we use is similar to stochastic gene parsing systems such as GenScan [8]. We give a formal definition of the model and describe how standard algorithms for evaluation and training can be adopted. Then we provide a brief overview of interpolated Markov chains which are used as the output distributions. Finally, we present the results of the SSMs both on the classification of a representative sequence set and on the scanning of large genomic sequences.

2 Methods

2.1 Stochastic segment models

Stochastic segment models (SSMs, see the paper of Ostendorf *et al.* [9] for an introduction and a comparison of different model types) have been proposed as a generalization of the widely used hidden Markov models (HMMs). Like HMMs, they consist of a set $\mathcal{Q}$ of connected states which can be characterized by an initial state distribution $\boldsymbol{\pi}$ and state transition distribution $\boldsymbol{A}$ with entries a_{ij}. Each state q_j contains an output distribution for the production of symbols which can be observed from the outside. While the output distribution of an HMM state can only emit a single symbol per state, each SSM state incorporates a joint distribution b_j which generates a sequence of symbols (a whole segment). The length of the generated segment underlies a duration distribution d_j associated with the state. Thus, the probability $P_j(\boldsymbol{w}_i)$ that a state produces a partial sequence $\boldsymbol{w}_i$ of length τ_i is given by

$$P_j(\boldsymbol{w}_i) = d_j(\tau_i) \cdot b_j(\boldsymbol{w}_i|\tau_i). \tag{1}$$

With a given valid segmentation $(\boldsymbol{s}, \boldsymbol{\tau}) = ((q_{s_1}, \tau_1) \ldots (q_{s_m}, \tau_m))$ of sequence $\boldsymbol{w}$ into segments $\boldsymbol{w}_j$, $(\sum_j \tau_j = |\boldsymbol{w}|)$, the probability of the sequence can be expressed as

$$P(\boldsymbol{w}, \boldsymbol{s}, \boldsymbol{\tau}) = \pi_{s_1} \prod_{i=1}^{m-1} P_{s_i}(\boldsymbol{w}_i) a_{s_i s_{i+1}} \cdot P_{s_m}(\boldsymbol{w}_m) \tag{2}$$

The output distribution b_j can itself be arbitrarily complex and take into account dependencies between the symbols within the segment. Depending on the field of application, different distributions such as Markov chains or HMMs may be suitable. Because the output distribution is conditioned on the duration, we have to provide either an individual distribution for each possible segment length or a mapping function from various segment lengths to a limited number, or the distributions have to be able to generate sequences of all valid lengths.

The idea of segment models is not new to the field of DNA sequence analysis – most gene finding systems which make use of stochastic models fit into the framework of SSMs. The GenScan system [8], in particular, uses a model structure similar to that proposed here (as a so-called *hidden semi-Markov model*). The difference is that we cannot expect the training material to be annotated in advance, which would allow for a supervised and individual learning of each output and duration distribution. For promoter regions we neither know how many segments we shall use for a successful recognition, nor have any means to separate all the segments from each other, because no promoter signal is guaranteed to occur in all sequences. This is opposed to the gene finding systems, where splice sites, for example, can be expected at the borders of exons and introns. A suitable algorithm for this task is described in the following section. A more elaborate description of our segment model formalism and implementation issues can be found elsewhere [10].

2.2 Algorithms for evaluation and training

The probability of generating sequence $\boldsymbol{w}$ with a segment model is equal to the sum of all possible segmentations over which the sequence can be uttered. Thus, using equation 2, we have

$$P(\boldsymbol{w}) = \sum_{\boldsymbol{s}} \sum_{\boldsymbol{\tau}} P(\boldsymbol{w}, \boldsymbol{s}, \boldsymbol{\tau}) \tag{3}$$

For HMMs, the corresponding probability can be computed efficiently by the *forward algorithm*. This algorithm calculates the forward variables $\alpha_{t,j}$ which contain the probability that the model is in state q_j at time t and has so far produced the symbol chain $w_1 \ldots w_t$. In HMMs, there is a state transition after each symbol, so the computation of $\alpha_{t+1,j}$ involves only the variables at time t. But for SSMs, the state duration is variable, so we have to sum up over all preceding variables where a state transition might have occurred. Therefore, we have to sum up over all possible segmentations $\boldsymbol{\tau}$. The resulting algorithm is depicted in figure 1.

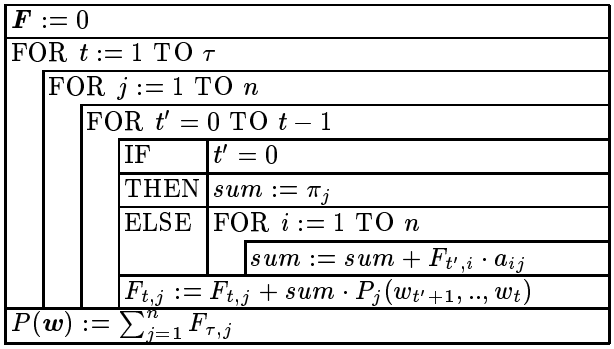

Figure 1: **Forward algorithm for segment models.** The input is $\boldsymbol{w} = w_1, .., w_\tau$. The matrix $\boldsymbol{F}$ contains the forward variables, n is the number of states. t is the actual time, j the actual state, and t' is the time where the state transition from state q_i to q_j takes place.

The evaluation of the forward algorithm involves many computations of the output distributions b_j, and has the consequence that we can make use of only those distributions that can be computed efficiently. One way to reduce the number of calculations drastically is to provide minimum and maximum durations τ_{min} and τ_{max} for the states, which is obviously application dependent. We will exploit this idea for the promoter model.

The most likely segmentation can be computed using a similarly adapted Viterbi algorithm, in which the sum over all possible segmentations is replaced by its maximum. Here, we use the Viterbi algorithm mainly inside a two-step training algorithm: First, we determine the most likely state sequence for each training sequence, then we treat this segmentation as the correct annotation. The resulting training material for each state is used to estimate the output and duration distribution. Of course, the probabilities of the state transitions and initial states are modified as well. The algorithm maximizes the Viterbi score of the model, i. e., the score obtained on the best segmentation is guaranteed to increase after each iteration. This so-called *Viterbi training* (see figure 2) usually results in a fast convergence.

2.3 Output and duration distributions

We already obtained promising results on the promoter recognition problem by the application of interpolated Markov chains [2], so we also used them as state output distributions. Here, we briefly revise the basic idea.

Given a sequence $\boldsymbol{w}$, the total joint probability can be computed with the chain rule:

Initialize model λ	
WHILE not converged or FOR a predefined number of cycles	
	$\hat{\boldsymbol{s}}, \hat{\boldsymbol{\tau}} := \text{argmax}_{\boldsymbol{s},\boldsymbol{\tau}} P_\lambda(\boldsymbol{s}, \boldsymbol{\tau}\|\boldsymbol{w})$ (Viterbi algorithm)
	$\forall i: \quad \bar{\pi}_i := \#(\hat{s}_1 = i)$
	$\forall i,j: \quad \bar{a}_{ij} := \#(\hat{s}_t = i \wedge \hat{s}_{t+1} = j)$
	$\forall i: \quad \hat{\pi}_i := \frac{\bar{\pi}_i}{\sum_i \bar{\pi}_i}$
	$\forall i,j: \quad \hat{a}_{ij} := \frac{\bar{a}_{ij}}{\sum_j \bar{a}_{ij}}$
	$\forall i: \quad$ Estimation of P_i including d_i and b_i
	$\lambda := (\hat{\boldsymbol{\pi}}; \hat{\boldsymbol{A}}; \hat{\boldsymbol{P}})$

Figure 2: **Viterbi training for segment models.** $\boldsymbol{A}$ denotes the state transition matrix with entries a_{ij}, $\boldsymbol{\pi}$ is the vector containing the start state probabilities. P_j is the segment density function for state j which incorporates the output and duration distribution. # is a function which counts the occurrence of its argument.

$$P(\boldsymbol{w}) = \prod_{i=1}^{|\boldsymbol{w}|} P(w_i | \underbrace{w_1 \ldots w_{i-1}}_{\text{context}}). \tag{4}$$

Because we cannot handle conditional probabilities with arbitrarily large context, we limit the size to $N-1$:

$$P(\boldsymbol{w}) \approx \prod_{i=1}^{|\boldsymbol{w}|} P(w_i | w_{i-N+1} \ldots w_{i-1}) \tag{5}$$

The resulting model is called a Markov chain (MC) of order $N-1$. The parameters of this model are the conditional probabilities on the right-hand side of equation 5. After a Maximum Likelihood parameter estimation, we obtain values for $\tilde{P}(v|\hat{\boldsymbol{v}})$ for each symbol v from the vocabulary $\mathcal{V}$ and each context $\hat{\boldsymbol{v}}$. To achieve more stable parameters, we compute the interpolation of all context lengths from 0 up to $N-1$ and use these as new parameter values. This can easily be done in a linear fashion:

$$\hat{P}(v|\hat{\boldsymbol{v}}) := \rho_0 \frac{1}{\mathcal{V}} + \rho_1 \tilde{P}(v) + \ldots + \rho_N \tilde{P}(v|\hat{\boldsymbol{v}}) \tag{6}$$

We used a more sophisticated approach which weights the individual parameters with their number of occurrence: Parameters which occur more frequent in the training material lead to a better statistics, and in this case we do not have to fall back to a shorter context as much as if the parameter seldom

occurs. Optimal interpolation coefficients ρ_i are calculated on a disjoint part of the training set using a gradient descent method [2].

Apart from the promising results, Markov chains are well suited out of a second reason. As we mentioned above, the evaluation of the output distributions must be calculated efficiently because of the large number of possible segmentations. With an MC, the total probability of a sequence can be broken down to single conditional probabilities per base, so we simply calculate these values along the whole sequence for each model state in advance and store them in a table. Thus, the calculation of a segment probability can be reduced to two table accesses and a subtraction, if we store the cumulative sum of the log probabilities.

As duration distributions, we simply use discrete distributions, represented as histograms of the relative frequencies. Because the Viterbi training only considers the most probable length, the values are smoothed with their left and right neighbours.

2.4 The promoter recognition system

The system for promoter detection in contiguous sequences contains a segment model for promoters and a model for non-promoters. The latter consists of two interpolated Markov chains, one trained on coding and one on intron sequences. They are treated as a mixture distribution with uniform weights.

For the application on contiguous sequences, we run a window of 300 bases over the sequence. Every 10 bases, we evaluate the window content with the promoter and the non-promoter model, and store the difference between the non-promoter and promoter scores. We obtain a curve describing the regulatory potential at each position. After a smoothing operation on the curve, a TSS is predicted at each minimum below a given threshold. The threshold is used to adjust the number of total predictions.

3 Data sets

We established representative sequence sets for the training and comparison of promoter recognition algorithms [2]. Currently, two sets of human and *D. melanogaster* sequences are available. These sets contain positive (promoters) as well as negative (introns and coding sequences) samples and are split in a number of subsets suited for cross-validation. The sets comprise a total number of 565 (265) promoters, 4345 (240) non-coding, and 890 (711) coding sequences (the numbers in parentheses are for the *Drosophila* set). The promoters contain 250 bases upstream and 50 bases downstream; the non-promoter

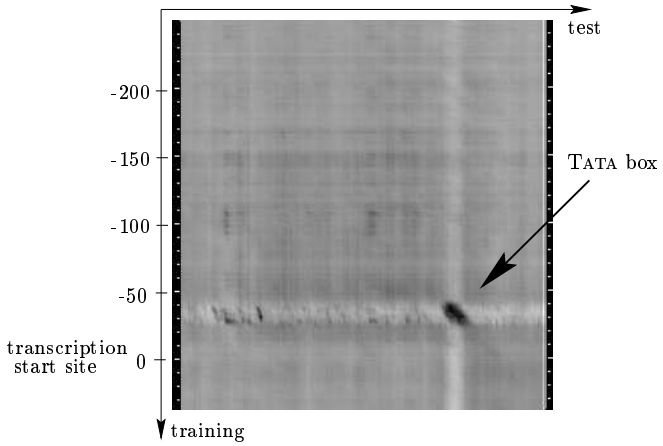

Figure 3: **Determination of initial model structure.** On the vertical axis, the starting position of the window on which a model was trained is given. The horizontal axis depicts how well the model trained on a certain window position performed in all windows. See the text for further explanation.

sequences are also 300 bases long. Further information and the sequences themselves can be retrieved via the Internet [a]. These sets will be referred to as "classification sets".

For the evaluation on contiguous sequences, we applied our human promoter model on the benchmark set of Fickett and Hatzigeorgiou [1]. It includes 18 vertebrate sequences with a total of 33,120 bp and contains 24 promoters. We are currently building a new reference set to pursue the evaluation in real-scale genomic regions, based on a contiguous 2.9 Mb sequence of *D. melanogaster* recently used for a community-wide genome annotation experiment [b].

4 Experiments and Results

4.1 Establishing a suitable model structure

To determine an initial promoter model structure, we performed the following experiment. We shifted a window of 12 bases along four-fifths of the human promoter sequences in the classification set. At each position, a fourth-order Markov chain was trained with the window content of all sequences. Markov

[a] http://www.fruitfly.org/seq_tools/human-datasets.html
[b] http://www.fruitfly.org/GASP1

Table 1: **Structure of the three-state promoter model.** Shown are the minimum and maximum length for each length distribution. The Markov chains used as output distributions are all of fifth order.

state	τ_{min}	τ_{max}
upstream	190	220
TATA	20	40
Inr/downstream	50	80

Table 2: **Structure of the five-state promoter model.** Shown are the minimum and maximum length for each length and the Markov order of each output distribution.

state	τ_{min}	τ_{max}	order
upstream	205	230	5
TATA	10	20	3
spacer	10	20	2
Inr	5	15	3
downstream	35	50	4

chains will be used as output distributions in our SSM, and the fourth order resembles the typical motif size of transcription elements. This model was then evaluated at every position of the remaining sequences, again within a window size of 12 bases. All the scores were summed up for each window, normalized and plotted against the position on which the window was trained (figure 3). High scoring windows appear in a dark color, and if dark regions appear on the diagonal, this indicates a position specific signal within a promoter region which can be detected by the model.

The only clearly visible position-specific signal is the TATA box region. Even at the TSS itself, there is no clear sign that the models trained on this region perform better than models trained on a different part of the promoter. This is somewhat surprising, but in accordance with the results of Zhang [7], who found that TATAAA is the only clear position specific six-tuple within promoters [c]. Obviously, the window size of 12 bases is too small to detect *region*-specific signals, such as transcription factor binding sites which occur more frequently in specific parts of the upstream region. We repeated the experiment with a window size of 50 bases, but this delivered no significantly different results. We thus decided to start our experiments with a three-state linearly connected model for upstream, TATA, and Inr/downstream region.

4.2 Performance on the classification data set

After a model structure was chosen, we performed a five-fold cross-validation experiment on the human classification set: We trained the models on four-fifths of the sequences with four cycles of Viterbi training which led to a good convergence. Then we evaluated them on the remaining part and averaged

[c] NB: A fourth-order Markov chain might be still too large to find a short TSS signal.

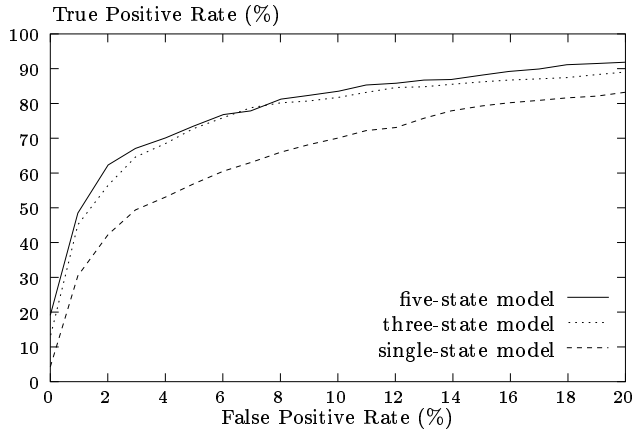

Figure 4: **Results on the human promoter classification set.** The receiver operating characteristics of a five-state, a three-state and a single-state promoter is depicted.

the results. We set upper and lower bounds τ_{min} and τ_{max} for the length distributions and initialized them with uniform values; as output distributions, we used fifth-order interpolated Markov chains. The model structure is given in table 1. The segment sizes are heuristic, but based on the experiment described above. The results were calculated with the forward algorithm instead of the Viterbi algorithm. This makes the probabilities comparable to the non-promoter model on a theoretically sound basis [d].

Figure 4 shows the resulting receiver operating characteristics (ROC), i. e. the recognition or true positive rate at different rates of false positives. The false positive rate can be adjusted by choosing different thresholds on the posterior probabilities of the concurring models. One can see immediately that the new promoter model with three segmental states performed much better than our previous system (one single state). This encouraged us to use three states for the core promoter: one for the TATA box, one for the initiator region around the transcription start site, and one for the spacer sequence between TATA box and initiator. Because these segments are smaller than the ones in the old model, we had less training material available for each state, so we chose smaller Markov orders for the output distributions to reduce the number of parameters. This should also lead to a better modeling of short signals such

[d] We also experimented with the Viterbi algorithm, but first runs on contiguous sequences showed that the output score (the difference between promoter and non-promoter model) was quite noisy, which lead to a large number of false predictions. Replacing the Viterbi score with the full probability calculated by the forward algorithm reduced this effect.

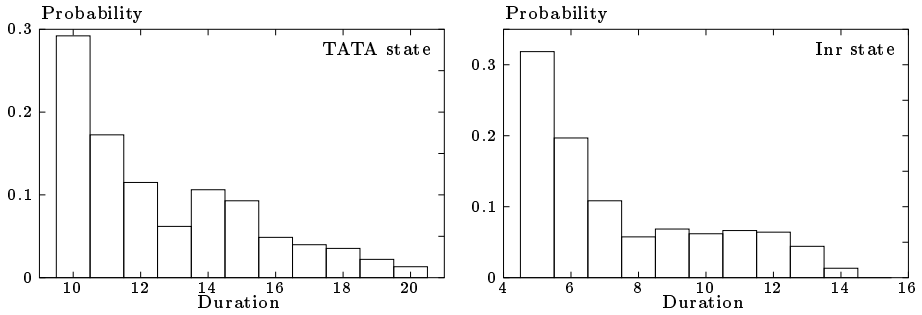

Figure 5: **Duration distributions for TATA and Inr state.** The distributions were initialised with uniform values and estimated with four cycles of the Viterbi training.

as the Inr. The new five-state model (table 2) is slightly better than the three-state, as can be seen in figure 4. The best averaged cross-correlation value (CC) is 0.66, at a false positive rate of 2 % and a true positive rate of 62.3 %. Compared with the single-state model, we were able to reduce the number of false predictions at the same recognition rate by more than two thirds. In figure 5, the learned duration distributions of the TATA and initiator state of one cross-validation experiment are depicted.

The same tests were also performed on the *D. melanogaster* sequence set. Figure 6 shows the results obtained with a five-state model with the same structure as the human one. The best CC is 0.68 at a rate of 7 % false positives and 75.4 % true positives.

4.3 Application on long genomic sequences

To see if we could obtain results for contiguous sequences as good as those for the classification set, we applied one model trained in the cross-validation experiments to search for the promoters in the genomic sequences from the survey of Fickett and Hatzigeorgiou [1]. We set the threshold at 2 % of false positives, where we obtained the best CC value.

We could detect 12 out of 24 promoters with a false positive rate of 1/895 bp. This is a slight improvement with respect to our previous system, where we detected the same number of promoters, but at a false positive rate of 1/849 bp. The system by Solovyev and Salamov [6], which was one of the best performing system in the survey, identified 10 promoters with a false positive every 789 bp.

We expected a better performance with the results from the previous sec-

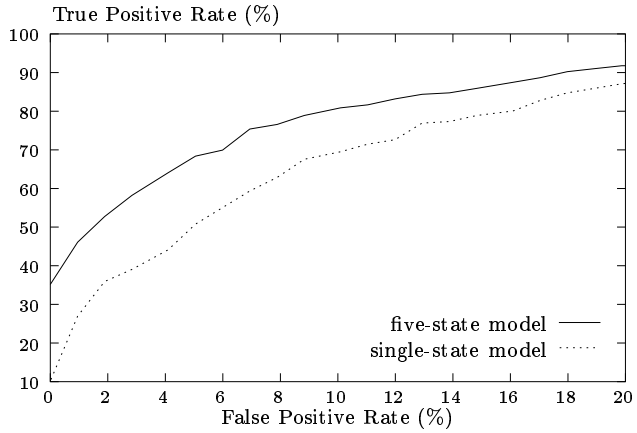

Figure 6: **Results on the *D.melanogaster* classification set.** The ROC curve of a five-state and a single-state promoter model are given.

tion in mind. Fickett and Hatzigeorgiou mention that the sequence set is not really representative, as the number of promoters is quite small. Furthermore, the test set was collected from articles which concentrated on transcriptional regulation, so the sequences might be biased towards special regulatory circumstances. Another explanation might be that the available training samples are not really representative. To clarify this, we aim at the evaluation of our models on a large and typical eukaryotic genomic sequence: the 2.9 Mb Adh region of *D. melanogaster* (mentioned in section 3) which contains approximately 230 genes. On a large data set, we can also study in detail the effect that the smoothing of the scores (see sec. 2.4) has on the overall performance.

5 Conclusions and Final Remarks

In this paper, we present a new approach for the stochastic modeling of eukaryotic polymerase II promoters, based on the general segmental structure of promoter regions. We could show a clear improvement of a five-state segment model on the classification of fixed-length sequences with respect to our previous approach, which modeled the promoter region as a whole. The results on genomic sequences are also improved, but not yet as much as we expected.

Currently, we have the following intention: to break up the linear structure of the model and introduce new states which run in parallel to others. Coupled with our Viterbi training algorithm, we aim to identify broad promoter clusters, depending on the optimal path chosen. Apart from better recognition, we

can obtain new insights by examining the parameters of the states. Such a model can also serve as a pre-classification step which enables data mining algorithms[11] to specifically search for significant transcription factor binding sites within the identified clusters.

The system can be accessed via the URL http://www5.informatik.uni-erlangen.de/HTML/English/Research/Promoter.

Acknowledgements

Uwe Ohler is funded by Boehringer Ingelheim Fonds and wishes to thank the people from the Berkeley Drosophila Genome Project for their constant support.

References

1. J. W. Fickett and A. G. Hatzigeorgiou. Eukaryotic promoter recognition. *Genome Res.*, 7:861–878, 1997.
2. U. Ohler, S. Harbeck, H. Niemann, E. Nöth, and M. G. Reese. Interpolated Markov chains for eukaryotic promoter recognition. *Bioinformatics*, 15(5):362–369, 1999.
3. R. D. Kornberg. RNA polymerase II transcription control. *Trends in Biochemical Sciences*, 21:325–326, 1996.
4. D. B. Nikolov and S. K. Burley. RNA polymerase II transcription initiation: A structural view. *Proc. Natl. Acad. Sci*, 94:15–22, 1997.
5. D. S. Prestridge. Predicting Pol II promoter sequences using transcription factor binding sites. *J. Mol. Biol.*, 249:923–932, 1995.
6. V. Solovyev and A. Salamov. The Gene-Finder computer tools for analysis of human and model organisms genome sequences. In *Proc. ISMB*, volume 5, pages 294–302, Menlo Park, 1997. AAAI Press.
7. M. Q. Zhang. Identifcation of human gene core promoters in silico. *Genome Res.*, 8:319–326, 1998.
8. C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.*, 268:78–94, 1997.
9. M. Ostendorf, V. Digalakis, and O. A. Kimball. From HMMs to Segment Models: a unified view of stochastic modeling for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 4:360–378, 1996.
10. G. Stemmer. Diploma thesis, University of Erlangen-Nuremberg, 1999.
11. A. Brazma, I. Jonassen, J. Vilo, and E. Ukkonen. Predicting gene regulatory elements in silico on a genomic scale. *Genome Res.*, 8(11):1202–1215, 1998.