From R. Hofestädt (Ed.), *Molekulare Bioinformatik*, Shaker, Aachen, 1998 (pp. 89–100).

DETECTION OF EUKARYOTIC PROMOTER REGIONS USING STOCHASTIC LANGUAGE MODELS

Uwe Ohler [1] and Martin G. Reese [2]

[1] Lehrstuhl für Mustererkennung (Informatik 5)
Universität Erlangen-Nürnberg
Martensstraße 3, D-91058 Erlangen
eMail: ohler@informatik.uni-erlangen.de
Phone: +49 (9131) 857775, Fax: +49 (9131) 303811

[2] Drosophila Genome Center
Department of Molecular and Cell Biology, 539 Life Sciences Addition
University of California at Berkeley, Berkeley, CA 94720, USA
eMail: mgreese@lbl.gov

ABSTRACT: We present a new *search-by-content* method to identify transcriptional regulatory regions in eukaryotic genomic sequences. The method is based on stochastic language models which are a straightforward generalization of oligomer statistics. We describe the theoretical background and different parameter estimation techniques used to build the models. The resulting language models are applied to classify fixed length sequences into the classes of promoters and non-promoters, and to search for transcription start sites in contiguous sequences. Detailed classification results for human and *Drosophila* data sets are presented, and the practical applicability of the models is demonstrated on an independent test set of vertebrate genomic sequences. On this set, which has already been used to compare different computational approaches for promoter recognition, the performance of our method is comparable to the best algorithms described so far. The number of false positives can be further reduced by a post processing step on the output scores. Examining both strands of the independent test set, the models thus are able to identify about half of the annotated transcription start sites (12 out of 22) while making a false prediction roughly every 800 base pairs.

1 INTRODUCTION

Computer based analysis of DNA primary sequences, especially of regulatory regions like promoters, is a challenging problem within the field of bioinformatics. As more and more large scale sequencing projects give rise to the number of long contiguous genomic sequences which may contain multiple genes, the need for computational methods to seperate these genes and find the exact location of the transcription start site and of the first exon is increasingly urgent. Methods which are available up to now still do not fulfill the requirements imposed on the prediction accuracy. This is mainly caused by the highly complex structure of eukaryotic polymerase II promoter sequences. Recent progress on the structure and function of polymerase II promoters is reviewed for example in (Kornberg, 1996) and (Roeder, 1996).

In a eukaryotic promoter, several signals are scattered within the region located near the transcription start site. These signals are binding sites of interacting transcription factors which regulate the expression of the particular gene by the eukaryotic polymerase II enzyme. The most prominent signals are certainly the signals within the core promoter region: The so-called TATA-box which is the binding site of the TFIID transcription factor that is involved in the interaction between DNA and eukaryotic polymerase II (Burley and Roeder, 1996), the initiator sequence at the start site itself (O'Shea-Greenfeld and Smale, 1992), and the downstream promoter element (Burke and Kadonaga, 1997). What makes promoter recognition difficult is the fact that signals other than those in the core region may occur in different numbers, spacing, and order. All of the occuring signals can be weakly conserved or missing altogether, showing a large dependency among each other. So, when searching for particular signals, one might miss a large number of weakly conserved promoters. On the other hand, the functionality of a signal seems to be determined not only by its sequence but also by the context in which it appears. Search methods are therefore plagued by a large number of false positives caused by single well conserved signals.

In (Fickett and Hatzigeorgiou, 1997) the available methods for promoter recognition were recently reviewed. On an independent test set the best methods were able to locate about 40–50 % of the true promoters with a false prediction every 500–800 base pairs. The methods reviewed included signal search methods, which either rely on the prominent signals around the transcription start site (Reese and Eeckman, 1998) or on a large collection of weight matrices for known transcription factors binding sites (Prestridge, 1995), as well as content search methods based on the statistics of short words (*oligomers*) which do not look for particular signals (Audic and Claverie, 1997; Hutchinson, 1996). In (Solovyev and Salamov, 1997) the two approaches are combined by judging a sequence using both weight matrices for the TATA-box and the initiator, and content based scores for the upstream region.

The method described here is a search-by-content method based on interpolated stochastic language models (SLMs). These models have been widely used for classification purposes in speech recognition (Jelinek, 1990; Kuhn et al., 1994; Schukat-Talamazzini et al., 1997). Interpolated SLMs can be regarded as a straightforward generalization of oligomer statistics, and we will show that they significantly outperform the oligomer approach.

2 METHODS

2.1 STOCHASTIC LANGUAGE MODELS

Recognising promoter sequences with a search-by-content method means to identify a whole regulatory region of fixed length within a contiguous genomic sequence. The position of the putative transcription start site is associated with a specific position within the region. The general approach thus is to slide a window of suitable length over the complete sequence in steps of a few bases and judge the sequence in the window. Therefore the problem can be broken down to the classification of a fixed length sequence $\mathbf{w} = w_1 \ldots w_T$, where each symbol w_i is taken from a finite vocabulary $\mathcal{V}$.

For simplicity, we first regard only one class of sequences. The probability $P(\mathbf{w})$ of the occurence of a particular sequence $\mathbf{w}$ can be written as follows, using the chain rule:

$$P(\mathbf{w}) = \prod_{i=1}^{T} P(w_i | w_1 \ldots w_{i-1}), \tag{1}$$

which means that one symbol in a sequence is dependent on all its predecessors, i. e. on the *history* of preceding symbols.

If we can establish a model which computes this probability, we have the means to determine how likely a sequence will occur in a specific class. A stochastic language model is exactly such a model which assigns a probability to a sequence of symbols.

2.2 ESTIMATION OF PARAMETERS

The right hand side of equation 1 contains a history of possibly infinite length which cannot be handled; therefore, an approximation is made by imposing a restriction on the history length. A possible approximation of the probability $P(\mathbf{w})$ is made by choosing the upper history length equal to $N-1$:

$$P(\mathbf{w}) \approx \prod_{i=1}^{T} P(w_i | w_{i-N+1} \dots w_{i-1}) \tag{2}$$

The resulting language model is called *N-gram* model; it is equivalent to the well-known oligomer approach with N as the length of the oligomer, which in turn is equivalent to a Markov chain of order $N-1$. We also refer to the parameter N as the considered *context*.

Using a training sample, the Maximum Likelihood estimation $\tilde{P}(w_i | w_{i-N+1} \dots w_{i-1})$ of the conditional probabilities with context N can be performed simply by counting:

$$\tilde{P}(w_i | w_{i-N+1} \dots w_{i-1}) = \frac{\#(w_{i-N+1} \dots w_i)}{\#(w_{i-N+1} \dots w_{i-1})}, \tag{3}$$

where # denotes the frequency of its argument in the training sample. Of course, one would like to choose a large context — the approximation made by a language model of higher order gets closer to the real probability as denoted in equation 1. Unfortunately, the number of parameters which have to be estimated increases exponentially with the number of N, and thus the ML estimates become far from being reliable because of the limited training sample size.

A compromise with respect to this trade-off between the model context and the training sample size can be made by introducing a weighted interpolation scheme.

2.3 INTERPOLATION METHODS

The basic idea of applying interpolation methods is to fall back on the probability estimation of subsequences shorter than N if the frequencies of the N-grams $\mathbf{v} = v_1 \dots v_N$ cannot be reliably estimated. An example is the *linear interpolation* between all the shorter subsequences up to the full length N:

$$\hat{P}(v_N | v_1 \dots v_{N-1}) = \rho_0 \frac{1}{L} + \rho_1 \tilde{P}(v_N) + \rho_2 \tilde{P}(v_N | v_{N-1}) + \dots + \rho_N \tilde{P}(v_N | v_1 \dots v_{N-1}) \tag{4}$$

The fraction $(1/L)$ accounts for unseen events and ensures that no probability is set to zero.

Equation 4 contains only one vector of interpolation coefficients, no matter if all the subsequences up to length N really occured in the training data. Obviously, the interpolation approach loses its eligibility if some of the summands $\tilde{P}(v_N | \cdot)$ with larger context are equal to zero because the training sample does not contain all possible N-grams. Indeed, this is likely to happen when a large number N is chosen and the available training data cannot account for the increasing amount of parameters. The solution lies in the estimation of different interpolation weights $\rho_i(\mathcal{H})$ depending on the available

length of the history $\mathcal{H}$. Such a model has the implicit effect of estimating N-grams of different context and is therefore called *polygram model* according to (Kuhn et al., 1994).

The interpolation weights $\rho_i(\mathcal{H})$ are optimized with respect to the Maximum Likelihood criterion by using an Expectation Maximization approach where the weights are regarded as hidden variables in a doubly stochastic process (see Schukat-Talamazzini et al., 1997). With this approach, the initial probabilities for the N-grams (equation 3) are reestimated using a second disjoint part of the training sample. Afterwards, a large weight will be assigned to those frequencies which can be reliably estimated; if only sparse data is at hand, the weights belonging to shorter subsequences will be increased.

Setting all the weights $\rho_0 \ldots \rho_{N-1}$ to zero and ρ_N to one results again in the well-known oligomer approach; the models with linear and polygram interpolation are a straightforward generalization of this approach combining oligomers of different length. The advantage of an interpolation scheme is that the model can take into account statistics of a higher order without running into the danger of overfitting the model to the training data.

A drawback of the linear and polygram interpolation scheme is certainly the inability to handle the particular N–grams individually. For example, the probability of an N–gram $\mathbf{x}$ of context six might be very large and the interpolation weight ρ_6 should therefore be large, whereas another 6-gram $\mathbf{y}$ occurs quite seldomly. By introducing an additional function $g_i(\mathbf{v}')$ which scores the reliability of the $(N-1)$–gram $\mathbf{v}' = v_1 \ldots v_{N-1}$ monotonically, the linear interpolation can be extended to handle this problem accurately:

$$\hat{P}(v_N|\mathbf{v}') = \frac{\sum_{i=0}^{N} \rho_i \cdot g_i(\mathbf{v}') \cdot \tilde{P}_i(v_N|\mathbf{v}')}{\sum_{i=0}^{N} \rho_i \cdot g_i(\mathbf{v}')}, \tag{5}$$

where $\tilde{P}_i(v_N|\mathbf{v}')$ denotes the frequency of $v_{N-i} \ldots v_N$. This interpolation scheme is called *rational interpolation* and uses only one vector of coefficients in contrast to the linear interpolation of polygrams which uses N vectors of increasing length $1 \ldots N$. The function $g_i(\mathbf{v}')$ is chosen to be a sigmoid funtion which is dependent of the frequency of the last i symbols of $\mathbf{v}'$:

$$g_i(\mathbf{v}') = \frac{\#_i(\mathbf{v}')}{\#_i(\mathbf{v}') + C} \tag{6}$$

In the case of $C = 0$, the function g_i is always equal to one and equation 5 becomes equivalent to the linear interpolation; we heuristically chose C to be equal to 10. In the rational case, the computation of optimal interpolation weights is carried out with a gradient descent algorithm instead of the EM approach which is used for linear coefficients. The details are omitted at this point and can be found in (Schukat-Talamazzini et al., 1997).

2.4 CLASSIFICATION OF A SEQUENCE

After a polygram model has been trained for each of the considered classes, the models can be used to classify a sequence. Let us assume that we have K classes $\Omega_1 \ldots \Omega_K$, and P_k denotes the language model for class k, $k \in \{1, \ldots, K\}$. Then we can compute the likelihood

$$P(\mathbf{w}|\Omega_k) = P_k = \prod_{t=1}^{T} \hat{P}(w_t|w_{t-N+1} \ldots w_{t-1}) \tag{7}$$

for each model and classify the sequence into sequence class $\hat{k}$ by computing the *a posteriori* probability using Bayes' rule:

$$\hat{k} = \arg\max_k P(\mathbf{w}, \Omega_k) = \arg\max_k (p_k \cdot P(\mathbf{w}|\Omega_k)) \tag{8}$$

As we have no exact knowledge about the *a priori* probabilities of our sequence classes, the values p_k are assumed to be uniformly distributed and are therefore neglected. Nevertheless, we are able to tune the models with respect to sensitivity and specificity. In this paper, the following approach is used: If only one class is of interest — i. e. we are only interested if the sequence is a promoter or not, no matter how many models we have trained for non-promoter sequences — we can compute the likelihood P_k for each class Ω_k with the language models and determine the difference between the score for the model of interest P_p and the best of the remaining models P_n. Including a length normalization, we obtain the following equation for the total score S:

$$S(\mathbf{w}) = \frac{P_n(\mathbf{w}) - P_p(\mathbf{w})}{len(\mathbf{w})} \tag{9}$$

In practice, the logarithms of the probabilities are used because of the more efficient computation and the prevention of numerically unstable values when regarding long sequences. In figure 1 an overview of the system structure discussed so far is given.

Choosing a suitable threshold value on the total score S, the selection of any percentage of false positives is feasible. The resulting curve of false positive rate vs. recognition rate over the whole range is called *receiver operating characteristic* (ROC) and gives us the full description of the performance of a classificator: When comparing two ROC curves resulting from different classificators, one can see instantly which one is better suited for the considered problem.

As a *single* number which also describes the performance of a classificator when considering two classes the *correlation coefficient* can be used:

$$CC = \frac{(TP \cdot TN) - (FN \cdot FP)}{\sqrt{(TP + FN) \cdot (TN + FP) \cdot (TP + FP) \cdot (TN + FN)}} \tag{10}$$

Herein, TP stands for true positives, TN for true negatives, FP for false positives, and FN for false negatives; these numbers denote the absolute numbers of correctly and wrongly classified sequences. The CC value lies within -1 and 1; the latter number occurs when a completely correct classification is available, a zero means arbitrary classification, and when all the classifications are wrong, the CC value is equal to -1. The correlation coefficient can be evaluated at the same corresponding values for recognition rate and false positive rate as the ROC curve; the best CC value then shows for which threshold the considered method gives the results deviating most from arbitrary classification. The CC value cannot substitute the complete ROC curves, but the measure is quite common and gives a first impression when comparing different models which were trained on the same data sets.

2.5 SEARCHING FOR PROMOTERS IN CONTIGUOUS SEQUENCES

To search for promoters in contiguous sequences by means of language models, we use a sliding window with a length of 300 bases. Every ten bases, the actual sequence in the window is classified as promoter or non-promoter. Because a whole promoter region is very likely to cause multiple predictions of several overlapping windows, a prediction is only made for each local minimum of the difference between non-promoter and promoter score which lies below the chosen threshold. The transcription start site is then assumed to be located at a specified position within the window; this location is used to evaluate the accuracy of the prediction.

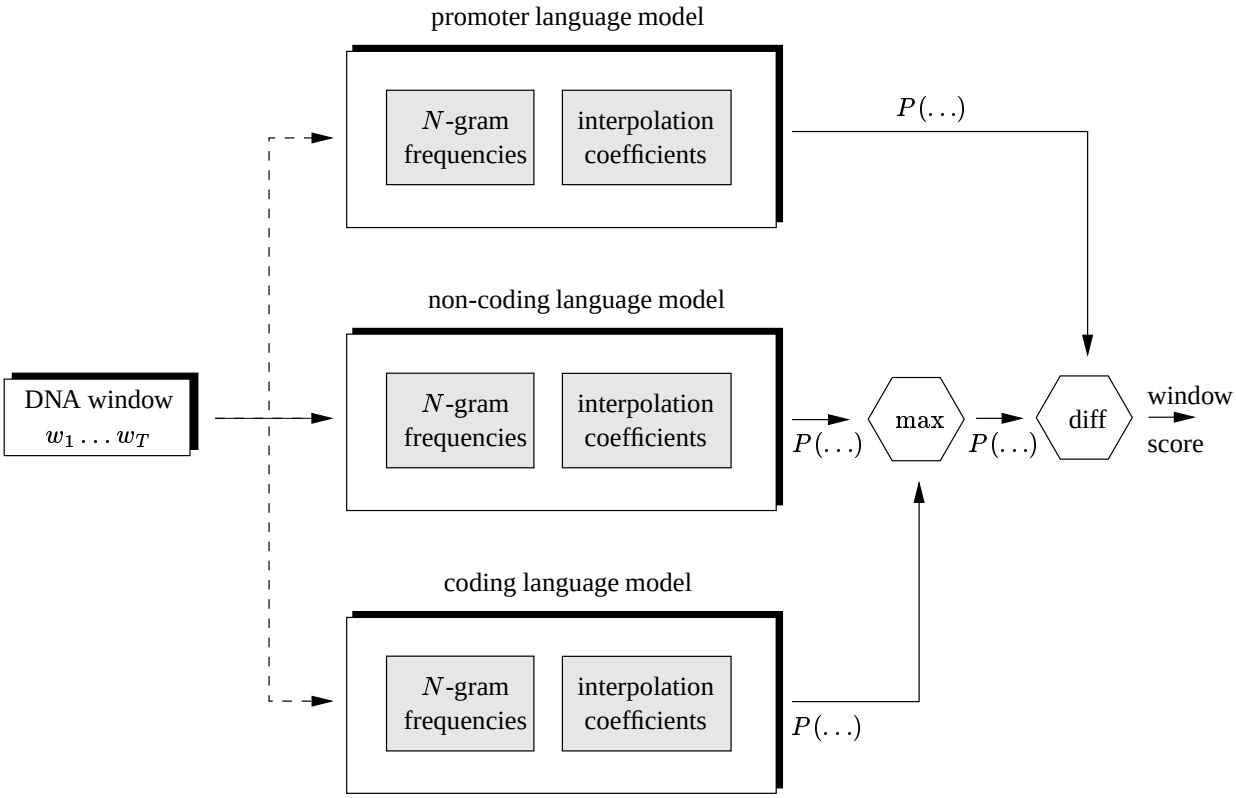

Figure 1: Promoter prediction with language models. A window of fixed length is shifted along the sequence, and every couple of bases the sequence in the window is judged by all the language models. The output is the difference between the best non-promoter and the promoter model score; the sequence of scores for the windows is then further processed as described in section 2.5.

To eliminate single false predictions, a post processing operation is applied on the score function S which is evaluated every ten bases over the whole sequence. By smoothing the resulting curve, single false promoter predictions as well as single non-promoter predictions within a promoter region are filtered out. We chose to apply the hysteresis threshold smoothing algorithm: A cursor of a chosen height is shifted over the curve from left to right, and the middle position of the cursor is always emitted as new output. As long as the next considered value lies within the cursor area, the cursor position is not moved vertically. If the next value lies above the cursor, it is moved up so that the upper rim corresponds with the value; if it lies below the cursor, it is moved down in an analogous way. With increasing cursor width, the curve is smoothed more and more.

3 DATA SETS

For training and evaluation of the methods, we used *D. melanogaster* and human data sets containing promoters, coding, and noncoding sequences. For the human promoter data set, we extracted all non-related vertebrate sequences except retroviruses from the Eukaryotic Promoter Database EPD rel. 50 (Perier et al., 1998). Retrieving only human promoter sequences from EPD would result in too small a dataset; EPD rel. 50 contained only 181 independent human sequences because each entry is rigorously validated and must be experimentally proven. Taking all vertebrate sequences instead is experimentally justified because vertebrate organisms share a reasonable amount of transcription

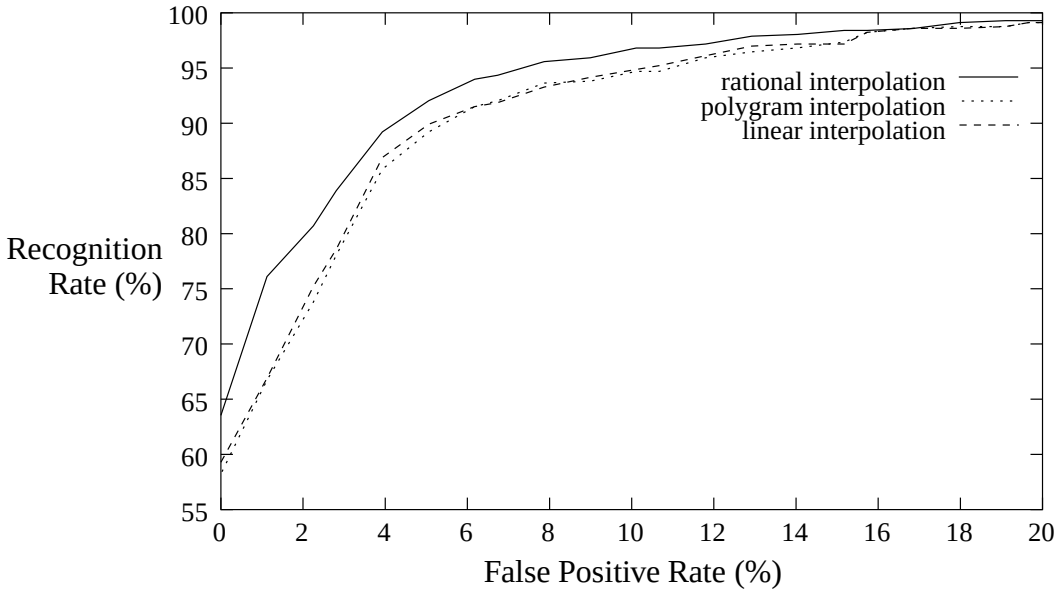

Figure 2: Receiver operating characteristics for different interpolation methods and context length seven, evaluated on the fixed length sequence set of promoters and coding sequences.

elements such as the well-known CAAT- and GC-boxes. Sequences with less than 40 bases upstream or 5 bases downstream from the annotated transcription start site were discarded to assure that at least the possible TATA-box and the initiator site were contained in each entry. This resulted in 565 entries, from which sequences up to 300 bases length (250 upstream and 50 downstream, if available) were extracted. As a *Drosophila* promoter set, we used the compilation of (Arkhipova, 1995), enriched by new sequences taken from EPD. Altogether, we had 256 *Drosophila* sequences with more than 40 bases upstream and 5 bases downstream.

For the coding and noncoding sequences, we used the exon and intron sequences of human and *Drosophila* genes contained in the data set for the GENIE genefinding system (Kulp et al., 1996). The exons were concatenated to form long coding sequences. Then, 300 bases long non-overlapping sequences were extracted. We divided the human data in five cross-validation sets containing 113 promoter, 180 coding, and 869 non-coding sequences each; the *Drosophila* data was divided in three sets, each comprising 85 promoters, 237 coding, and 80 non-coding sequences. In the remaining text, these sets will be referred to as ”sequences of fixed length” and can be obtained from the authors upon request to make a thorough comparison with other methods possible.

To evaluate the performance of the system on long contiguous sequences, we made use of the independent data set in (Fickett and Hatzigeorgiou, 1997). Using this data set we additionally have the possibility to compare the system’s behaviour with other programs aimed at promoter recognition. The original data set consists of 18 mammalian sequences containing 24 annotated and experimentally proven promoters with a total of 33,120 bp, from which 17 sequences — two of them enriched by new flanking sequences — could be retrieved; those 17 sequences had a total of 34,284 bp and contained 22 promoters. None of these sequences was contained in the training set. The evaluation on the contiguous sequences was carried out on both strands; recognition results are therefore given in base pairs instead of single bases.

false positives (%)	**recognized promoters (%)**		
	promoter vs. CDS	*promoter vs. intron*	*promoter vs. CDS/intron*
0.0	63.5 (0.72)	12.6 (0.33)	6.9 (0.24)
1.0	76.1 (0.80)	22.9 (0.48)	34.5 (0.49)
2.0	80.7 (0.82)	44.1 (0.53)	43.9 (0.52)
3.0	83.9 (0.83)	**49.2 (0.53)**	**50.4 (0.53)**
4.0	89.2 (0.86)	51.3 (0.51)	54.2 (0.52)
5.0	**92.0 (0.87)**	53.3 (0.50)	59.1 (0.53)
6.0	94.0 (0.87)	57.5 (0.50)	61.7 (0.52)
7.0	94.3 (0.87)	60.0 (0.50)	63.7 (0.51)
8.0	95.6 (0.87)	63.5 (0.51)	66.2 (0.50)
9.0	95.9 (0.86)	64.6 (0.49)	68.3 (0.50)

Table 1: Promoter classification on vertebrate sequences with stochastic language models using rational interpolation and context length seven. For a certain percentage of false positives, the corresponding cross-validated recognition rate and the correlation coefficient is given. The recognition rate with the highest correlation coefficient is printed in bold (CDS = coding sequence).

4 RESULTS AND DISCUSSION

As a first step, we determined experimentally which N–gram context and interpolation method were best suited for promoter recognition. We applied N–grams with different context lengths using the three interpolation methods on the fixed length sequence set, comprising human promoters and coding sequences. Four of the five parts were used as training and one part as an independent test set. Figure 2 shows a part of the receiver operating characteristics obtained on this set using language models with context length seven for which we could obtain the best results. The figure shows clearly that rational interpolation outperforms the linear and polygram approaches, and that no improvement could be achieved using polygram instead of linear interpolation; nevertheless, even the linear scheme shows very good results. With increasing context length, no considerable improvement was achieved. We therefore applied a careful cross-validation experiment on the fixed length sequence set, using models trained with a context of seven and rational interpolation. Table 1 shows the recognition rate on the sets of fixed length sequences for three discrimination tasks: promoter vs. coding sequences, promoter vs. intron sequences, and promoters vs. both coding and non-coding sequences. Here, the depicted numbers were obtained by averaging the results of five experiments; in each experiment the model was trained on four parts of the sequence data, leaving one part out at a time and evaluating the performance on the part not used for training.

The discrimination performance between promoters and coding regions is stunning; at a false positive rate of 5 % already 92 % of the promoter sequences were classified correctly (correlation coefficient 0.87). Nevertheless it is also very clear that a classification between promoter and introns is much more difficult — the best CC value obtained was 0.53, at a false positive rate of 3 % and a recognition rate of 49.2 %. Most probably this stems from the much weaker information contained in the introns compared to the strong coding information of the exons. On applying models on the three-part set of promoters, non-coding, and coding sequences, the results are comparable to the two-class problem of promoters and coding sequences, resulting from the much larger sample size of intronic sequences.

Corresponding results obtained by making a threefold cross-validation on the set of promoters,

false positives (%)	**recognized promoters (%)**		
	promoter vs. CDS	*promoter vs. intron*	*promoter vs. CDS/intron*
0.0	49.5 (0.63)	9.0 (0.21)	10.15 (0.29)
1.0	66.3 (0.75)	18.0 (0.28)	27.0 (0.43)
2.0	75.3 (0.78)	23.8 (0.31)	35.9 (0.49)
3.0	82.8 (0.82)	23.8 (0.31)	39.1 (0.48)
4.0	86.3 (0.83)	30.5 (0.35)	44.1 (0.49)
5.0	**93.0 (0.86)**	35.2 (0.37)	50.8 (0.53)
10.0	96.9 (0.81)	45.3 (0.39)	**69.5 (0.59)**
15.0	97.3 (0.76)	49.6 (0.37)	78.9 (0.58)
20.0	97.7 (0.70)	60.2 (0.41)	87.1 (0.58)
25.0	98.0 (0.65)	**67.6 (0.43)**	90.6 (0.55)

Table 2: Promoter classification on *D. melanogaster* sequences with stochastic language models using rational interpolation and context length six. For a certain percentage of false positives, the corresponding cross-validated recognition rate and the correlation coefficient is given. The recognition rate with the highest correlation coefficient is printed in bold (CDS = coding sequence).

coding, and noncoding regions of *D. melanogaster* can be seen in table 2. Here, again rational interpolation, but with a smaller context of six proved to deliver the best recognition rate. Obtaining the best result with a smaller context is most probably a consequence of the more limited training sample size. One can see that the overall performance is somewhat comparable to those on human sequences, but especially the discrimination between non-coding sequences and promoters is worse. This is possibly due to the more than twentyfold larger sample of non-coding sequences which was available for the training of the human model. Because of the much smaller average intron size in *Drosophila* genes, the effect of the poor promoter vs. non-coding discrimination on the overall performance is not as strong as for human sequences.

We compared the performance of our language models with the oligomer statistics — that is a stochastic language model without any interpolation. For this purpose we trained one model using hexamer statistics and another SLM with rational interpolation and context length six and evaluated both on one cross-validation experiment for the promoter vs. coding sequence recognition task. The obtained results show that, using the available limited data, the hexamer model cannot be as reliably trained as the interpolated models: At a false positive rate of 1 %, the model with rational interpolation can recognise over ten percent more promoter sequences than the hexamer model. This clearly shows that the SLMs are able to estimate the statistics much better whereas the hexamers are overfitting to the sparse training data and perform much worse on unseen test sequences.

Finally, we applied one model trained on promoters, coding, and non-coding sequences to the task of finding promoter regions in longer vertebrate DNA sequences, using the set of contiguous sequences cited in the promoter prediction program survey of (Fickett and Hatzigeorgiou, 1997). In this survey, a prediction is judged as correct if an annotated transcription start site lies within 200 bases downstream and 100 bases upstream from the predicted site. That means that a prediction is correct as long as it is made somewhere within a large part of the regulatory region. Using this criterion, we could detect 12 out of 22 (54.5 %) of the promoters while having one false prediction on average every 504 base pairs. The two programs which achieved the best performance in the survey could detect 54 % and 42 % of the promoters with a false positive rate of 1/460 bp and 1/789 bp, respectively (Reese and Eeckman, 1998; Solovyev and Salamov, 1997). These numbers show that the

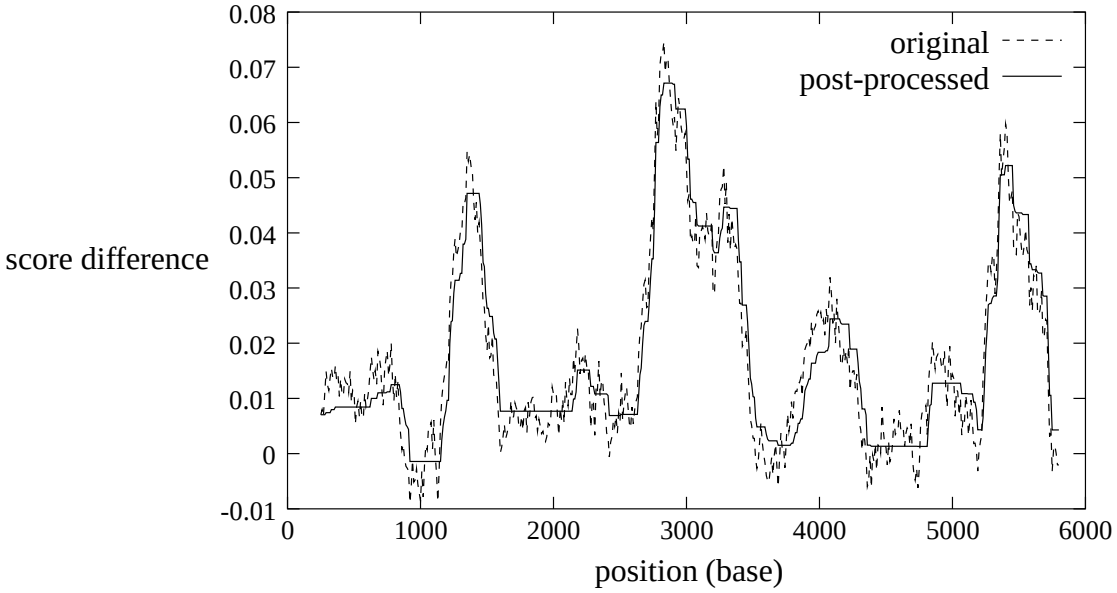

Figure 3: Output of the system on a large contiguous sequence, before and after applying the automated post processing step. The system makes three predicitons, one for each local minimum below a certain threshold. The predictions are located at positions 920, 3690, and 4740; the annotated transcription start site is located at position 935.

performance of the polygram models is comparable to the best available tools for promoter prediction, but the number of test sequences is far too small to make a fair and exact comparison possible.

The results were further improved by smoothing the output score function with the hysteresis algorithm as described in section 2.5. By applying this post-processing step with a heuristically chosen cursor width of 0.015, the number of false predictions could be drastically reduced to one every 797 bp while making no forfeits on the recognized promoters. An example of the performance on one human test sequence, the human phenol sulfotransferase gene (GenBank accession code HSU54701), is shown in figure 3. In this example, within a sequence of about 6,000 bases length three predictions are made, one of which is located close to the annotated transcription start site.

5 CONCLUSIONS

In this paper we showed that stochastic language models can be successfully used to discriminate between promoter and non-promoter (coding and non-coding) regions. The application on contiguous sequences achieves promising results — after applying a postprocessing step, our models give recognition results comparable to the best methods described in the recent survey of (Fickett and Hatzigeorgiou, 1997): We identified the transcription start sites of 12 out of 22 promoters, with a false prediction every 800 bp. Two of the non-identified transcription start sites were located very close to the sequence start, so that no complete 300 base pair long sequence — the length our models were trained on — was available. One of the eight remaining promoters missed was also not detected by any of the nine programs evaluated in (Fickett and Hatzigeorgiou, 1997). At the moment, we have no non-promoter model for intergenic sequences; if a reliable training sample for this sequence class can be obtained, the performance is likely to improve because of the more accurate sequence modeling. Obtaining such a sample though is difficult; most database entries contain only single genes, and additional sequence parts outside the gene sequence are not definitely annotated.

The prediction accuracy of the TSS location was quite good despite the fact that SLMs do not use location specific information. From the 12 promoters recognized, no prediction was further than 100 bases away from the annotated TSS, and seven of them were made within 20 bases from the true start site.

Compared to oligomer statistics which can be seen as a special case, the power of approaches using interpolated statistics lies in a much stabler parameter estimation, especially when only sparse data is at hand as is actually the case for promoter sequences. In the authors' opinion, this is the reason for the considerable improvement made by SLMs on the promoter recognition task. With a recognition rate as high as the one which can now be achieved, one can think about an integration of a promoter recognition module in a large-scale gene finding system like GENIE (Kulp et al., 1996). Thereby, the modules for gene structure determination and promoter recognition are likely to benefit from each other: On the one hand, the large number of false promoter predictions may be reduced because of the absence of an exon region following it; on the other hand, an improved accuracy on the detection of the first exon is yielded, and the separation of multigenic sequences into single genes becomes possible. An exact promoter prediction can also be useful to identify the correct full length cDNA.

We also combined the search-by-content method of this paper with the search-by-signal method of the time delay neural networks as described in (Reese and Eeckman, 1998). Preliminary results show that on the set of fixed length sequences, the number of false predictions made by the neural network alone could be reduced by half when applying both methods on the sequences and combining the scores appropriately. This exploitation of different knowledge will be the topic of future research.

This work was partially supported by a grant of the Boehringer Ingelheim Fonds to Uwe Ohler.

References

I. Arkhipova. Promoter elements in drosophila melanogaster revealed by sequence analysis. *Genetics*, 139:1359–1369, 1995.

S. Audic and J.-M. Claverie. Detection of eukaryotic promoters using Markov transition matrices. *Computers and Chemistry*, 21(4):223–227, 1997.

T. W. Burke and J. T. Kadonaga. The downstream core promoter element, DPE, is conserved from Drosophila to humans and is recognized by TAFII60 of Drosophila. *Genes Dev*, 11:3020–3031, 1997.

S. K. Burley and R. G. Roeder. Biochemistry and structural biology of transcription factor IID (TFIID). *Annu Rev Biochem*, 65:769–799, 1996.

J. W. Fickett and A. G. Hatzigeorgiou. Eukaryotic promoter recognition. *Genome Research*, 7:861–878, 1997.

G. B. Hutchinson. The prediction of vertebrate promoter regions using differential hexamer frequency analysis. *Comp. Appl. Biosc.*, 12(5):391–398, 1996.

F. Jelinek. Self–organized Language Modeling for Speech Recognition. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*, pages 450–506. Morgan Kaufmann, San Mateo, 1990.

R. D. Kornberg. RNA polymerase II transcription control. *Trends in Biochemical Sciences*, 21: 325–326, 1996.

T. Kuhn, H. Niemann, and E. G. Schukat-Talamazzini. Ergodic hidden Markov models and polygrams for language modeling. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 1, pages 357–360, 1994.

D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman. A generalized hidden Markov model for the recognition of human genes in DNA. In *Proc. Fourth Int. Conf. Intelligent Systems in Molecular Biology*, St. Louis, 1996.

A. O'Shea-Greenfeld and S. T. Smale. Roles of TATA and initiator elements in determining the start site location and direction of RNA polymerase II transcription. *J Biol Chem*, 267(2):1391–1402, 1992.

R. C. Perier, T. Junier, and P. Bucher. The Eukaryotic Promoter Database EPD. *Nuc. Ac. Res.*, 26(1): 353–357, 1998.

D. S. Prestridge. Predicting Pol II promoter sequences using transcription factor binding sites. *J. Mol. Biol.*, 249:923–932, 1995.

M. G. Reese and F. H. Eeckman. Time-delay neural networks for eukaryotic promoter prediction, 1998. submitted.

R. G. Roeder. The role of general initiaton factors in transcription by RNA polymerase II. *Trends in Biochemical Sciences*, 21:327–334, 1996.

E. G. Schukat-Talamazzini, F. Gallwitz, S. Harbeck, and V. Warnke. Rational Interpolation of Maximum Likelihood Predictors in Stochastic Language Modeling. In *Proc. European Conf. on Speech Communication and Technology*, pages 2731–2734, Rhodes, Greece, 1997.

V. Solovyev and A. Salamov. The Gene-Finder computer tools for analysis of human and model organisms genome sequences. In *Proc. Fifth Int. Conf. Intelligent Systems in Molecular Biology*, pages 294–302. AAAI Press, 1997.