Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 01-030

Promoter Prediction of Prokaryotes

Michihiro Kuramochi, Mukund Deshpande, George Karypis, Qing Zhang, and Vivek Kapur

July 23, 2001

PROMOTER PREDICTION FOR PROKARYOTES [a]

Michihiro Kuramochi, Mukund Deshpand, George Karypis
Department of Computer Science/Army HPC Research Center
University of Minnesota
4-192 EE/CS Building, 200 Union St SE
Minneapolis, MN 55455

Qing Zhang
Department of Veterinary PathoBiology
University of Minnesota
1971 Commonwealth Ave
St. Paul, MN 55108

Vivek Kapur
Biomedical Genomics Center
University of Minnesota
1971 Commonwealth Ave
St. Paul, MN 55108

The availability of computational methods to identify and define the precise structure and location of promoters in prokaryotic genomes will provide a critical first step towards understanding the mechanisms by which genes are organized and regulated. We examine three different methods for promoter identification, two of which are adopted from related work and the other is a novel approach based on feature extraction. By the results of a set of experiments we evaluated prediction accuracy for identifying promoter regions from non-coding regions.

1 Introduction

Whole-genome sequencing and analysis of organisms provides an unprecedented opportunity to discover new genes and possible mechanisms by which they may be regulated. Ever since the complete genome sequencing of the first free-living organism in 1995, the bacterium *Haemophilus influenzae* [1], it has become abundantly clear that there is a very large number of putative or predicted genes which were previously unrecognized. In many prokaryotes, including those that have been extensively studied for decades, this number often approaches 25% or more [2]. A common means of identifying these new genes is

[a] This work was supported by NSF CCR-9972519, EIA-9986042, ACI-9982274, by Army Research Office contract DA/DAAG55-98-1-0441, by the DOE ASCI program, and by Army High Performance Computing Research Center contract number DAAH04-95-C-0008. Access to computing facilities was provided by the Minnesota Supercomputing Institute.

through the application of various gene-finding algorithms [3,4,5,6,7,8] of varying sophistication that predict transcriptional start and stop sites. While there has been considerable progress in identification of new genes through the application of these algorithms, the discovery of how these genes are regulated is not as advanced and detailed mechanisms of gene regulation and co-regulation on a genome scale are lacking. The first step by which genes are regulated is by modulation of the level of transcription. This occurs when the enzyme, RNA polymerase, which is responsible for catalyzing the production of the new RNA transcript, recognizes and binds a specific sequence of DNA that is most often located upstream of the gene and known as the promoter element. It is therefore the location and primary structure of the promoter that determines the precise site at which the new RNA transcript will be initiated and the abundance of the nascent transcript.

Several decades of investigation on the structure and function of prokaryotic promoters have led to the recognition of three major features [9,10]: (i) a 35 box; (ii) a 10 box; and (iii) the presence of a purine nucleotide (either Adenine or Guanine) at the transcription initiation site. The 35 and 10 boxes are so named for their approximate location prior to the transcription initiation site. They are typically hexamer sequences with the primary structure of TTGAGA for the 35 element and TATAAT for the 10 one. However, it is important to recognize that whilst these canonical sequences have been described, there is considerable variation in both the structure and the relative location of these elements, making their exact identification using computational methods a non-trivial task.

The availability of computational methods to identify and define the precise structure and location of promoters in prokaryotic genomes will provide a critical first step towards understanding the mechanisms by which genes are organized and regulated. To achieve this goal, we studied three methods, two of which were adopted from previous work, and the other is a novel approach for identification of promoter elements in the genome of the model prokaryote, *Escherichia coli*. Overall, the results of our investigations show that (i) frequency-based method and Markov chain models are inaccurate and (ii) incorporation of feature-based scheme improves the accuracy considerably to 92.5% for up to 5000 base long upstream regions.

2 Related Research

The majority of the research in promoter identification has been focused on developing such computational methods for eukaryotic organisms [11,12,13,14], and there exists limited recent research in developing techniques for promoter identification in prokaryotic organisms.

The early research in promoter identification was primarily focused on trying to differentiate a promoter element from a coding region [15]. However, as highly accurate algorithms have been developed for gene finding in prokaryotic organisms [3,4], this type of promoter identification techniques are now of little use. One of the most recent published work on developing methods for predicting prokaryotic promoters was done by Thieffry *et al.* [16]. Their study was focused on the *E.coli* genome and they used pattern matching techniques to identify new promoter elements similar to those that are already known. Their experiments showed that such techniques achieve 75% precision with high accuracy; however, their overall experimental evaluation was confusing.

3 Methods

Although predicting promoter locations in a prokaryote genome does not provide any important information [11] about the location of genes, it is computationally difficult to predict precise locations of promoters [16]. This is because it is feasible to precisely identify coding regions with help of computational approaches [3,4]. Thus, the task remained to be solved is to distinguish promoter regions from non-promoter upstream regions.

In this study, we examine three different approaches for predicting the location of promoters. The first is motivated by a similar set of techniques that were developed for eukaryotic genomes and can work in an unsupervised setting. The second approach uses Markov chain techniques, to build a model of the known promoters that can be used to classify if a particular location is part of a promoter. The last approach selects a set of features that encode sequential aspects of the dataset and then uses the selected features for classification. Both the Markov model and the feature extraction techniques can only work in a supervised setting (*i.e.*, there has to exist some promoters that are already known).

3.1 Frequency-based Approach

Several frequency-based approaches have been proposed for eukaryotic genomes and were shown to produce reasonably good results [12,13,17]. The driving principle of these methods is that if there exists a short sequence of nucleotides that occurs more frequently in the upstream region of a gene than in the coding region, then this short sequence must have been conserved for a reason, and is a good candidate for being a promoter.

In particular these algorithms work as follows. First, for each gene they select a fixed size sequence upstream from the gene's translation start site. Then, they randomly select subsequences from the coding regions of the different genes so that they will end up selecting roughly the same number of nucleotides as those present in all the selected upstream regions. Given these two sets of sequences, these algorithms then find all the subsequences that satisfy a certain minimum frequency constraint σ, *i.e.*, each of discovered subsequences is present at least σ times in the dataset. The length of the subsequences that are being discovered is usually limited to be around seven to eight bases. Now, for each such pattern they compute its *specificity* to the upstream region, which is the ratio of the number of times it appears in an upstream region divided by the number of times it appears in a coding region. Finally, these algorithms use a threshold ρ on the specificity of a pattern, and prune all the patterns that have a specificity below ρ. The patterns that remain are then considered to be the promoter elements.

This basic algorithm can been improved in two different ways. First, instead of counting the frequency of exact patterns, the patterns themselves can be relaxed to include one or more wild card characters, allowing us to model near matches [12]. Second, algorithms have been developed for determining the upstream specificity of a pattern without comparing it against its frequency to the coding region [18]. However, the basic idea of these approaches remains the same, in the sense that a pattern specific to the upstream regions are predicted as promoters.

One of the main advantage of the frequency-based approaches is that it only requires that we know the location of gene's coding and upstream regions. As a results, these approaches can be used without any *a priori* knowledge of known promoter elements, and can be used to predict putative promoters even in not-well-studied genomes.

3.2 Markov Chains

Markov chains are well-known probabilistic technique for classifying sequential data [19], and they have been successfully used in a number of sequence-based prediction tasks, such as gene identification in prokaryotes. Unlike the frequency-based approach, Markov chain is a supervised learning algorithm; thus, it requires training datasets to build models and to make prediction.

For each of the classes that we are trying to classify, Markov chains first build a model from sequential training inputs. The fundamental assumption of Markov chains is that an event in a sequential stream depends only on a limited number of preceding events. In the case of DNA sequence, we can rephrase it that the probability of an occurrence of each type of a nucleotide n at a particular location x_i is determined by its preceding sequence of $x_{i-k}x_{i-k+1}\cdots x_{i-1}$. The number of preceding bases that influence x_i is called the *order* of Markov chains. For example, in the first-order Markov chains, only a single forerunner x_{i-1} affects the choice of a nucleotide at x_i, and in the second-order only a pair of bases $x_{i-2}x_{i-1}$ determines the probability of the choice of a nucleotide at x_i. A resulting model is a table of transitional probabilities from a prefix of certain length to an incoming character.

During classification, the Markov chains of the different classes are used to compute the probability that a particular sequence is being generated by the learned model. The model that has the highest probably is used to classify the new sequence.

For the prokaryotic promoter prediction problem, we first build two Markov chain models, one for promoter regions and the other for non-promoter upstream regions. Then, for each location of an input sequence, we compute the two probabilities, the probability that the given input sequence of length L is a promoter, and the one that the sequence is *not* a promoter. For example with the first-order model, the probabilities are $P(C = \text{promoter} \mid x_{i-1}x_i)$ and $P(C \neq \text{promoter} \mid x_{i-1}x_i)$ and we take the class which has higher probability and assign it as prediction.

3.3 Feature-based Approach

Our third approach for developing algorithms for promoter prediction was motivated by our recent work on developing scalable clustering algorithms for protein sequences [20]. In that work, we first used sequential pattern discovery algorithms to find all frequently occurring subsequences in the database, we then projected each protein into a new space that had these features as its dimensions, and then we used traditional vector-space clustering algorithms to find the clusters. Our experimental results showed that even though each protein was eventually represented as a multi-dimensional vector in which the ordering relations between the different subsequences was completely lost, the overall clustering solutions were extremely good.

In the *feature-based* approach we follow a similar framework. The basic idea is the following. First, create a multi-dimensional space containing 4^w dimensions, each dimension corresponding to one of the 4^w possible w-mers. Then, take the upstream region of each gene and break it into overlapping segments of a certain length n. Finally, take each one these segments and represent it as a vector in the 4^w-dimensional space, by *subscribing* it to all the possible w-mers that it contains. As a result of this transformation, the underlying upstream regions are now represented using traditional multi-variable vectors. We can assign a class to each of these vectors based on whether or not they are derived from the known promoter regions. In our algorithm, sequence segments that were mostly (but not entirely) obtained from the known promoter regions were assigned to the positive class, whereas the rest of the sequences were placed in the negative class.

Given such a representation of the data set, any traditional machine learning algorithm can be used to learn a model that differentiates one class from the other. For our experiments we decided to use support vector machines (SVM)[21,22], as they have been shown to consistently produce superior results than other classifiers.

4 Experimental Results

We experimentally evaluated the performance of the various algorithms for promoter prediction on the *E.coli* genome.

We obtained 471 locations of transcriptional start sites of *E.coli* publicly available from the PromEC[23] database. Out of those 471, 54 promoters are discarded because they are completely covered by coding regions. Since the internal structure of those promoters are not yet determined completely, we take 101 bases as a *promoter region* from -75 to $+25$ relative to the transcriptional start site and applied the three different methods which are described in Section 3.

By combining the predictions with the actual classes we can partition the test data into four classes. The *true positives* and the *true negatives* which are correctly predicted to be part of the positive or negative class, respectively; and the *false positives* and *false negatives* which are the test data that were incorrectly predicted as positives or negatives, respectively. A common way of measuring that performance is to use two measures called the *precision* and *recall*. The precision p of a binary classifier is defined as

$$p = \frac{N_{\text{true positives}}}{N_{\text{true positives}} + N_{\text{false positives}}},$$

and the recall is defined as

$$r = \frac{N_{\text{true positives}}}{N_{\text{true positives}} + N_{\text{false negatives}}}.$$

The precision measures what fraction of upstream regions that are predicted positive are actually of promoters, and the recall measures what fraction of the true promoters are actually predicted as positive. An alternate way of evaluating the performance of a classifier is to look at its accuracy, which is defined as the fraction of correct predictions. However, when the different classes are of significantly different sizes, the accuracy measure can be misleading, and looking at precision and recall provides more meaningful information.

4.1 Frequency-based Approach

To evaluate the performance of the frequency-based approaches to correctly identify the known promoters of *E.coli*, we performed the following experiments. We selected the genes whose upstream regions did not overlap with coding regions of the opposite strand. This gave us a total of 962 upstream regions, out of which 101 had known promoters. Then we randomly selected an equal number of genes and computed all oligonucleotide sequences that satisfied certain support constraints up to length 10, allowing up to 2 wild cards. For each of these frequently occurring oligonucleotides we computed their upstream specificity and filtered out patterns whose specificity was lower than a certain threshold ρ. The patterns that were left were used to predict promoter regions. In calculating the specificity we took into account of the different number of total nucleotides in the coding and non-coding regions.

Table 1 summarized the results obtained by this approach using different values of σ and ρ. The precision and recall figures were computed by looking at the number of nucleotides of the known promoter elements that were predicted correctly or incorrectly. A set of nucleotides is predicted correctly if it matches one of the selected patterns.

Based on the scheme described in Section 3.1, we extracted frequent and upstream-specific patterns as shown in Table 1 to predict promoters. We tested frequency threshold $\sigma = 10$ and 20, and the ratio threshold $\rho = 2$, 5 and 7.

Table 1: Prediction results by frequency-based approach for known promoters in upstreams. Those upstream regions do not have any overlap with coding regions on the other strand. The column labelled with "Patterns" shows the number of extracted patterns of up to 10 bases satisfying the frequency σ and the ratio ρ thresholds. A pattern may contain 2 wild cards at most.

Frequency σ	Ratio ρ	Patterns	Precision (%)	Recall (%)
10	2	121386	37	48
20	2	40247	37	48
20	5	5709	41	47
20	7	2013	45	44

Looking at these results, we can see that we increase ρ from 2 to 7, the precision increases at the cost of the decrease of recall. However, the quality of the overall results was quite low.

4.2 Markov Chains

In this section we describe the methodology for a Markov model based classifier. For each gene whose promoter element is known we took an upstream sequence of n bases in length from the transcription start site. If the promoter element was included in these n bases, that particular upstream region was kept in the set, otherwise it was removed. In our experiments we let n take the values of 200, 500, 1000, 2000 and 5000 leading to five different datasets. Each of the datasets was used to evaluate the performance of Markov chains using a 10-fold cross validation. That is, we split the data into 10 parts, 9 to be used for training and 1 for testing and the overall experiments was repeated 10 times, each time with a different part for testing.

As discussed in Section 3.2, during the training we learn the transitional probabilities for the positive class (*i.e.*, the known promoter regions) and the transitional probabilities for the negative class (*i.e.*, the rest of the selected upstream regions). For classification we use a sliding window of w bases long and the log-ratio of the two probabilities is used to predict whether or not each nucleotide is a part of a promoter.

Table 2 shows the results for the different values of n and w. Note that we built separate models for the two strands as we found that some of the transitional probabilities differ substantially. Looking at these results we can see that the best performance was obtained for $n = 200$. Depending on the parameters, the classifier was able to achieve precision and recall in the range of 70–80%. However as n increased the quality of the predictions degraded substantially. Also note that the second-order Markov chain actually leads to worse results than the first order model. Looking at the sensitivity with respect to w, we can see that as w increases the results tend to get better.

4.3 Feature-based Approach

The performance of the feature-based approach was evaluated using the same data sets as those used for the Markov model-based scheme. However, because these two approaches model the data in a different way, the definition of the positive and negative class, as well as the classification and evaluation methodology was somewhat different.

In particular, the transactions were created using the sliding window approach discussed in Section 3.3, by using a window of length w and sliding it k bases each time.

We first explain the procedure for generating examples, which will be used for both training and testing the classifier. We take an inter-genic sequences of base pairs and break this sequence into shorter overlapping sequences of length

Table 2: Prediction results by the first and the second order Markov chain models

Order (k)	Window Size (w)	Maximum Upstream Length (n)	Forward Strand Precision (%)	Forward Strand Recall (%)	Backward Strand Precision (%)	Backward Strand Recall (%)
1	10	200	66.5	66.4	71.3	72.9
		500	52.4	58.5	53.2	60.4
		1000	40.2	52.9	33.4	42.0
		2000	27.0	55.4	24.4	43.1
		5000	17.5	56.7	11.0	43.0
	20	200	66.5	73.7	71.3	81.2
		500	54.0	60.9	52.7	63.6
		1000	43.1	54.7	33.1	40.7
		2000	30.5	56.4	25.4	41.8
		5000	20.0	58.0	12.1	41.6
	40	200	68.3	81.6	68.2	88.1
		500	57.9	64.0	52.7	69.2
		1000	45.9	54.5	32.9	40.0
		2000	35.5	57.1	25.6	39.7
		5000	23.6	59.2	13.0	39.4
2	10	200	68.7	61.5	70.4	67.4
		500	50.7	55.3	51.1	53.2
		1000	40.1	51.3	32.5	42.7
		2000	27.3	54.5	25.1	44.6
		5000	17.9	57.0	11.4	44.6
	20	200	69.5	67.7	70.0	74.4
		500	51.8	55.5	50.6	56.1
		1000	41.3	50.1	32.5	41.2
		2000	30.9	55.0	25.1	41.5
		5000	20.6	58.5	12.6	42.4
	40	200	70.9	73.6	67.3	81.8
		500	54.2	57.3	50.0	60.4
		1000	42.1	47.9	32.2	40.5
		2000	35.2	54.4	24.6	39.5
		5000	24.2	60.3	13.4	39.8

$w = 50$. The amount of the overlap is equal to $w - k$. These short overlapping sequences obtained from inter-genic sequences make up the examples in the dataset. The class label of this sequences is computed by finding the class label of majority of base pairs making up the sequence. Thus, for example if the majority of the base pairs making up the sequence are promoters then the example sequence will be labeled as a promoter sequence and similarly for non-promoter sequences. Sequences having length less than $w = 50$ or sequences that do not have a clear majority are ignored [b] After obtaining the example sequence the complete dataset is split into two parts, training sequences and testing sequences.

Next we describe the methodology for building classification models and the motivation behind our approach. We first identify set of features and then transform example sequences into this feature space. After doing this transformation each example is represented as a boolean feature vector. Depending on the presence and absence of features the indices in the boolean vector are set to true or false. These boolean feature vectors are then used for building the classification model.

The features used for classification are made up of four consecutive base pairs (*i.e.*, $w = 4$), hence enumerating all possible base pairs (of length four) gives us a feature space of size $4^w = 256$. To identify the features present in an example sequence, we slide a window of size 4 across the example sequence, each time incrementing its position by one till the window slides across the entire sequence. Each position of this window (of size 4) constitutes a feature which is supported by the example. This special arrangement of sliding window means that each example consists of exactly $47 (= 50 - 4 + 1)$ features.

Table 3 shows the performance achieved by the feature-based approach using SVM as a classifier. The tables shows the results obtained for the five datasets, for different values for the length of the sliding window and the length of the oligonucleotides that were used to derive the various features. These results correspond to those obtained after 10-way cross validation. Note that as it was the case with the Markov model-based technique, we build separate classifiers for the two different strands.

Looking at the results in the table we can see that the feature-based approach produces significantly better results than those produced by any of the earlier schemes. Its precision and recall even for the large data sets is over 92%. The other thing to note is that the overall quality seems to decrease for the data sets obtained by looking at upstream regions of size 1000, and 2000,

[b] We use the value of 66% to signify clear majority, *i.e.*, if 2/3 of the $w = 50$ base pairs making up the example sequence are promoters then the sequence will be classified as a promoter.

but it improves when we consider regions of length 5000. One of the reasons for that may be that the additional promoters that were included as we moved from 2000 to 5000 are easier to classify. However, this is something that we are currently investigating.

Table 3: Prediction results of the feature-based approach. Those results are obtained by 10-way cross validation. Precision shows the value of precision and recall at the break-even point. Columns labelled with "Inter-genic Regions" and "Known Promoters" show the number of non-coding regions and the number of the known promoters included in the dataset, depending on the value of the maximum upstream limit.

Maximum Upstream Length	Forward Strand			Reverse Strand		
	Inter-genic Regions	Known Promoters	Precision (%)	Inter-genic Regions	Known Promoters	Precision (%)
200	34	40	97.5	41	44	97.8
500	64	81	91.3	72	95	91.3
1000	79	103	88.6	92	120	86.1
2000	104	132	88.6	106	138	82.5
5000	129	166	92.9	141	176	92.5

5 Conclusion

We examined the prediction performance of two previously proposed schemes for promoter prediction, the frequency-based approach and Markov chains, by performing cross-validated experiments using the precision and recall measure. We also proposed another novel method, the feature-based approach with the SVM classifier and evaluated its prediction accuracy as well. The experimental result showed the feature-based approach achieved higher prediction accuracy (in terms of both precision and recall) than that of the other two methods.

References

1. R. D. Fleischmann, M. D. Adams, O. White, R. A. Clayton, E. F. Kirkness, and A. R. Kerlavage et al. Whole-genome random sequencing and assembly of haemophilus influenzae rd. *Science*, 269:496–512, 1995.
2. C. M. Fraser, J. Eisen, R. D. Fleischmann, K. A. Ketchum, and S. Peterson. Comparative genomics and understanding of microbial biology. *Emerg Infect Dis*, 6:505–512, 2000.
3. A. L. Delcher, D. Harmon, S. Kasif, O. White, and S. L. Salzberg. Improved microbial gene identification with glimmer. *Nucleic Acids Research*, 27:4636–4641, 1998.

4. S. Salzberg, A. Delcher, S. Kasif, and O. White. Microbial gene identification using interpolated markov models. *Nucleic Acids Research*, 26:544–548, 1998.
5. D. Frishman, A. Mironov, and M. Gelfand. Starts of bacterial genes: estimating the reliability of computer predictions. *Gene*, 234:257–265, 1999.
6. D. Frishman, A. Mironov, H. W. Mewes, and M. Gelfand. Combining diverse evidence for gene recognition in completely sequenced bacterial genomes. *Nucl. Acids Res.*, 26:2941–2947, 1999.
7. W. Hayes and M. Borodovsky. How to interpret anonymous genome? machine learning approach to gene identification. *Nucleic Acids Research*, 8:1154–1171, 1998.
8. A. V. Lukashin and M. Borodovsky. GeneMark.HMM: new solutions for gene finding. *Nucleic Acids Research*, 26:1107–1115, 1998.
9. Benjamin Lewin. *Gene VII.* Oxford University Press, Cambridge, UK, 2000.
10. S. Busby and R. H. Ebright. Promoter structure, promoter recognition, and transcription activation in prokaryotes. *Cell*, 79:743–746, 1994.
11. James W. Fickett and Artemis G. Hatzigeorgiou. Eukaryotic promoter recoginition. *Genome Research*, 7:861–878, 1997.
12. Alvis Brazma, Inge Jonassen, Jaak Vilo, and Esko Ukkonen. Predicting gene regulatory elements in silico on a genomic scale. *Genome Research*, 8(11):1202–1215, November 1998.
13. Harmen J. Bussemaker, Hao Li, and Eric D. Siggia. Regulatory element detection using correlation with expression. *Nature Genetics*, 27, Feburary 2001.
14. Uwe Ohler, Stefan Harbeck, Heinrich Niemann, Elmar Nöth, and Martin G. Reese. Interpolated markov chains for eukaryotic promoter recognition. *Bioinformatics*, 15(5):362–369, 1999.
15. Paul B. Horton and Minoru Kanehisa. An assessment of neural netowrk and statistical approaches for prediction of e.coli promoter sites. *Nucleic Acids Research*, 20(16):4331–4338, 1992.
16. Denis Thieffry, Heladia Salgado, Araceli M. Huerta, and Julio Collado-Vides. Prediction of transcriptional regulatory sites in the complete genome sequence of escherichia coli k-12. *Bioinformatics*, 14(5):391–400, 1998.
17. J. van Helden, B. Andre, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *Journal of Molecular Biology*, 281:827–842, 1998.

18. H. J. Bussemaker, H. Li, and E. D. Siggia. Building a dictionary for genomes: identification of presumptive regulatory sites by statistical analysis. *Proc. Natl. Acad. Sci. USA*, 97:10096–10100, 2000.
19. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids.* Cambridge University Press, Cambridge, UK, 1998.
20. Valerie Guralnik and George Karypis. A scalable algorithm for clustering protein sequences. In *BIOKDD01: Workshop on Data MIning in Bioinformatics*, 2001.
21. V. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, New York, 1995.
22. V. Vapnik. *Statistical Learning Theory.* John Wiley, New York, 1998.
23. Ruti Hershberg, Gill Bejerano, Alberto Santos-Zavaleta, and Hanah Margalit. PromEC: An updated database of escherichia coli mrna promoters with experimentally identified transcriptional start sites. *Nucleic Acids Research*, 29(1), 2001.