Nucleic Acids Research, Vol. 20, No. 16 4331–4338

An assessment of neural network and statistical approaches for prediction of *E.coli* promoter sites

Paul B.Horton and Minoru Kanehisa*
Institute for Chemical Research, Kyoto University, Uji, Kyoto 611, Japan

Received March 11, 1992; Revised and Accepted July 15, 1992

ABSTRACT

We have constructed a perceptron type neural network for *E.coli* promoter prediction and improved its ability to generalize with a new technique for selecting the sequence features shown during training. We have also reconstructed five previous prediction methods and compared the effectiveness of those methods and our neural network. Surprisingly, the simple statistical method of Mulligan *et al.* performed the best amongst the previous methods. Our neural network was comparable to Mulligan's method when false positives were kept low and better than Mulligan's method when false negatives were kept low. We also showed the correlation between the prediction rates of neural networks achieved by previous researchers and the information content of their data sets.

INTRODUCTION

With the expanding sequence data available and the abundance of possible techniques for sequence analysis, it is important to evaluate and compare the effectiveness of diverse methods for predicting functional sites in nucleic acid sequences. The oldest and simplest technique is to construct a consensus sequence from known sites and compare it against candidate sites. This is effective for well defined sites such as restriction enzyme recognition sites but does not allow for hierarchical base preferences. For less well defined sites, base frequency matrices containing the frequency of each base at each position in a compilation of known sites are commonly used. This preserves much more information than a consensus sequence alone and has been used frequently in the analysis of promoters (1–3). Numerous attempts have been made to use pattern recognition techniques to go beyond a simple frequency table or to interpret a frequency table in a sophisticated way (4–7). One particular pattern recognition technique that has been reported to give good results is the use of neural networks (8–11).

We have trained a neural network to predict promoter sites and introduce a technique for selecting input to the network, in which the network itself determines the input units to be used during training. The networks used have no hidden units but are shown some higher order information, in the form of the base content of certain regions of input sequences, thus allowing for limited higher order learning. Our network is then compared with five previous methods, as well as previous neural networks, for prediction accuracy using common data sets.

We demonstrate the importance of using consistent methodology when evaluating the efficacy of alternative prediction schemes. It is quite common to quote one percentage and claim to have improved on previous results when in actuality one percentage does not indicate much at all. To do a fair job of evaluation the percentage of both false positives and false negatives must be calculated at several representative thresholds and the generality or relative difficulty of the test set used must be considered. In fact, it is often the case that the test set contains sequences that are quite similar to training sequences. We found that the differences in reported prediction accuracies using the same method could be explained in terms of the information content of the data sets used for estimation.

METHODS

Database

The promoter sequences used were obtained from Harley and Reynolds' compilation (12). Sequences less than full length (14 bases before the −35 hexamer and 11 bases after the −10 hexamer) were discarded. This also served to remove all of the heat shock promoter elements from the database. To avoid introducing biases by counting variations of the same basic sequence more that once, sequences that contained nearly identical sections were grouped together (Table 1). Nearly identical was defined here as having a stretch of 15 continuous bases with one or no mismatches. The corresponding stretches were required to occur within 3 bases of the same position in the sequence. The resulting 147 groups were equally weighted during both testing and training regardless of the number of members within each group. All of the groups consisted of 1,2,3 or 6 sequences allowing the groups to be given equal weight by including single sequences in the data set 6 times, pairs of sequences 3 times, triples twice and sextuples once. Thus 100 groups would correspond to a data set size of 600 (non-independent) sequences. Sequences from coding regions in GenBank release 65.0 were chosen randomly as negative data.

* To whom correspondence should be addressed

Training and Test Sets

In order to provide an objective test of methods that were based on the earlier promoter compilation of Hawley and McClure (13), the groups were divided into 98 groups that were included in Hawley's compilation and 49 newer groups. The 98 older groups (training set B) were divided into a 49 group training set (training set A) and a 49 group test set (test set A) to be used for experimenting with network architectures and parameters. The 49 newer groups then provided a test set (test set B) for final evaluation of predictive algorithms. Hawley and McClure's alignment was used for the earlier groups, while Harley and Reynolds' alignment was used for the later ones. Coding region data was grouped into three sets, none of which shared any identical sequences with any other. The coding training set contained 5146 sequences which were trained for 21 continuous window positions allowing for approximately 110,000 rounds of training without repeating negative examples. As with the promoter data, two negative test sets were made, negative test set A for experimenting with parameters (5807 sequences) and negative test set B for final evaluation (11614 sequences). The negative test sets were evaluated at only one window position per sequence.

Information Content

The information content of a set of aligned sequences was calculated by

$$\text{Information Content} = \sum_{\text{all bp}} \left(2 - \sum_{b=A}^{T} f_b \cdot \log_2(f_b)\right)$$

where bp stands for the base positions of the sequence and f_b is the frequency of each base at the given base position. This is equivalent to Schneider *et al*'s (14) formula using 0.25 for all *a priori* base frequencies.

Neural Networks

We developed several programs in C with a UNIX workstation to create and analyze neural networks. All of the networks discussed here were without hidden units (perceptron architecture).

Initial Input. As shown in Figure 1, two regions of the sequences were selected to be used as possible input, an upstream region of 25 bases with the start of the −35 hexamer being in the 15th position and a downstream region of 27 bases with the start of the −10 hexamer being in the 11th position. Spacing data from 15−21 bases was input with binary coding to seven input units. During training of promoter examples the spacing from the respective compilation's alignment was shown to the network directly, while at all other stages the spacing that gave the highest network output was used. The network output should ideally be one for promoters and zero for non-promoters, therefore training negative examples with the alignment giving the highest output is equivalent to training the network with the negative alignment on which it performed the worst.

Data Representation. Input was shown to the network in the form of individual bases and also in the form of the combination of two bases joined by an OR function, giving a total of seven possible input units for each base position. For example, if the given base was a purine the input unit for (A or G) was turned

Table 1. Training and Test Data.

Training Data[a]

434PR; 434PRM; alaS; ampC; araBAD; araC; argCBH; aroH; bioA; bioB; bioP98; cat; CloDFrnaI,pBRRNAI; ColE1-P1; ColE1-P2; deo-P1,deo-P2; fdVIII; fd x; fol; gap-P1,gal-P2; g-s-tnpA; g-s-tnpR; glnS; his; hisA; hisJ(St); ilvGEDA; lacI; lacP1; lacP115; lacP2; l17; lcin; lL57; lPI; lPL,lPR; lPo; lPR′; lPRE; lPRM; leultRNA,rrnABP1,rrnD-P1,rrnE-P1,rrnG-P1; rrnX1; lex; lpp; M1rna; malEFG; malK; malT; P22ant; P22mnt; P22PR; P22PRM; pBR322bla; pBR322P4; pBR322primer; pBR322tet; pBRP1; fXA; fXB; fXD; pori-l; pori-r; R100rna3; R100RNAI; R100RNAII,R1RNAII; recA; rplJ; rpoA; rpoB; rrnABP2,rrnG-P2; rrnDEXP2; RSFprimer; RSFrnaI; S10; spc; spot42r; str; supB-E; T7-A1; T7-A3,T7-A2; T7-C; T7-D; thr; Tn10Pin; Tn10Pout; Tn10tetA; Tn10tetR; Tn5IR; Tn5neo; tnaA; trp; trpP2; trpR; trpS; tufB; tyrT; uvrB-P1; uvrB-P2; uvrB-P3.

Test Data[b]

argE-P1; argE-P2,argE/LL13; argF; aroF; aroG; carAB-P1; carAB-P2; CloDFcloacin; colE1-C; crp; cya; dapD; deo-P3; divE; dnaA-1p; dnaA-2p; dnaK-P1; FplstraM; fumA; glnL; glyA/geneX; gnd; hisS; htpR-P1; htpR-P3; ilvIHP4; ISlins PL; livJ; malPQ,malPQ/Pp16,malPQ/Pp18; manA; metA P1; metA P2; micF; MuPe; NR1rnaC; NR1rnaC/m; ppc; pyrB1-P1; pyrE-P1; rpmH1p; rpmH1p; rpmH2p; rpmH3p; sdh-P2; ssb; sucAB; Tn266abla-Pa; trxA; tyrT/109; uncI; uvrD.

a) The training data is from Hawley and McClure's compilation using their alignment but Harley and Reynold's nomenclature was used when possible. b) The test data is from Harley and Reynold's compilation using their alignment. The sequences separated by commas were very similar to each other and therefore grouped together within their data set (see text). New sequences from Harley and Reynold's compilation that were very similar to sequences within the training data were not included in the test data.

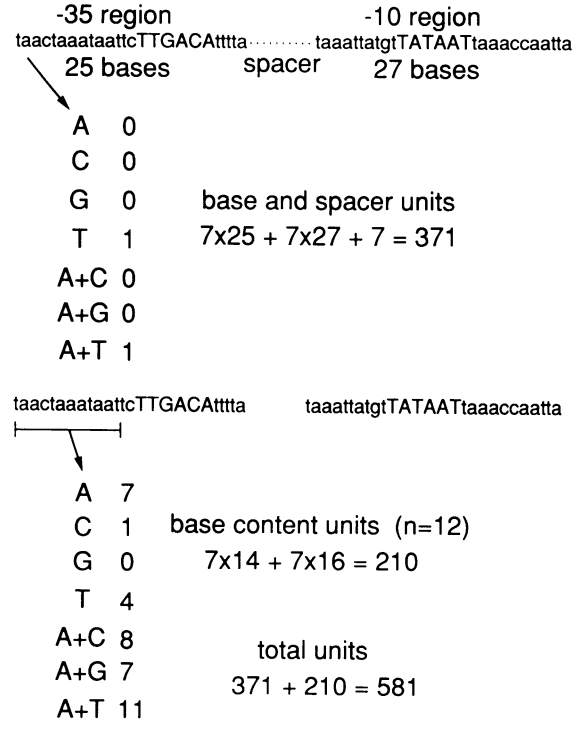

Figure 1. The type and number of input units are shown for a full network with 12 base length content units in addition to individual base units. Note that in the actual networks the input was normalized (see text).

on, thus allowing the information, that a purine base is important, to be learned by the network in a concise way. The content data for length n regions of the sequences were also input. This should facilitate the learning of patterns such as a low G+C content or the presence of poly-A subsequences. In the final network only single position units and length n=12 content units were used (Fig. 1).

Trimming the input units. The network input units were 'trimmed' by removing the input that was used the least (i.e. whose weight had the smallest absolute value) after a fixed number of rounds of training, and then retraining the network as shown in Figure 2. This process was repeated until an appropriate amount of input units remained. In order to compare weights representing individual bases to those representing the base content of a window of length n it was necessary to normalize the inputs. For normalization we used the following function which gives a range of (0,1) and an expected value of 0.5 for any length n:

$$\text{Input} = \sum_{m=0}^{k-1} \binom{n}{m} p^m q^{n-m} + \tfrac{1}{2}\binom{n}{k} p^k q^{n-k}$$

Here k is the number of times the given base or pair of bases occurs in the input window of length n; p is the *a priori* probability of occurrence, equal to 1/4 for bases and 1/2 for pairs; and q is 1−p. As can be seen, the summation term is equal to the cumulative probability of the given base being found less than k times in a binomial distribution.

Training. The networks were trained with the back propagation algorithm (15) with a learning rate of 15 divided by the number of inputs, corresponding to 0.185 for 81 input units and a momentum of 0.5. As in Rumelhart *et al.* (15) the momentum was implemented as follows:

$$\Delta W_i(n+1) = (\eta)(\delta)(i) + (\alpha)\Delta W_i(n)$$

where $\Delta W_i(n+1)$ is the change to the weight of the i^{th} input on the $n+1^{th}$ round of training, η is the learning rate, δ is the error signal, i is the value of the i^{th} input and α is the momentum.

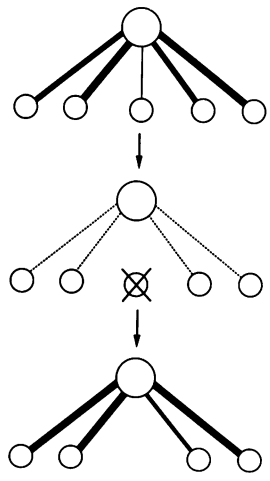

Figure 2. The process of trimming one input unit is shown. The input unit that corresponds to the smallest (absolute value) weight after training is trimmed, then all the weights are set to zero and the network is retrained without the trimmed input unit.

The error signal is the difference between the output of the network and the target output for the given input. During some of our preliminary work the target output was set to one for promoters and zero for non-promoter sequences. It was found, however, that setting the target values to 0.9 and 0.1 as in (11) improved generalization ability somewhat and those target values were used for all final results. A complete description of the back propagation algorithm has been described elsewhere (11,15−17) and therefore will not be described here.

Determining Parameters. The optimal learning parameters were determined using the accuracy of the network on test set A. Here the network's accuracy was calculated as the percent of negative examples classified correctly when the network threshhold was adjusted for 80% correct classification of the test set A promoters. If for example, 80% of the test set A promoters obtained a score of 0.6 or more with a particular set of network weights then that network's accuracy was considered to be the percent of negative test set A sequences that scored below 0.6. In particular, the optimal number of rounds of training was determined by sampling the accuracy of the network with test set A while using training set A to train the network. Thus the optimal number of rounds of training does not directly reflect the difficulty of the training examples (e.g. training the network until the total error falls below a certain level), but is empirically derived using test set A in order to reduce overlearning. This is necessary because sometimes (as in the trimmed input curve of Fig. 3) a neural network's performance against unseen data actually goes down when the network is trained past a certain optimal number of rounds of training. The final network was trained with all of the older data (training set B) with the learning parameters and number of rounds of training established using only the older data (training and test set A).

Previous Methods

Mulligan et al.'s Method. Mulligan *et al.*'s weight matrix was used directly (2). Their method is essentially equivalent to summing the entry of the base frequency matrix/sqrt(4) over the length of the candidate sequence, and then adding the frequency of the spacing class/sqrt(7) to obtain a score. This score is then compared to a threshold to make the desired prediction. Here 4 and 7 correspond to the number of possible bases and the number of possible spacing classes.

Staden's Method. Staden's published matrix was used directly for comparison (3). His method is similar to Mulligan *et al.*'s except that the log of the frequencies is summed without dividing by the square root. Summing the log of the frequencies is equivalent to multiplying the frequencies, which can be thought of as probabilities.

Alexandrov and Mironov's Method. Aexandrov's published matrix and another matrix obtained from Alexandrov personally were used (7). Their method is a modified 'general portrait' method which finds the distinguishing vector giving the largest separation between two sets of points in a feature space. They also strove to find the minimum set of variables needed for successful prediction and their method does, in fact, have fewer parameters than any other listed here. It should be noted that they used some of the promoters in Harley and Reynolds' compilation thus excluding a true test set for evaluation.

O'Neill's Method from the Journal of Biological Chemistry. This method was reproduced from O'Neill's paper (5). To reproduce his data base an additional promoter sequence was taken from Gentz and Bujard (18). This prediction scheme consists of a series of tests in which a candidate sequence is compared to a template and passes that test if it has more than a certain number of matches to the template. Separate sets of tests are provided for bacterial spacing classes 16,17,18 and the phage 17 base spacing class. If a candidate sequence passes any of those sets of tests it is predicted to be a promoter.

O'Neill's Method from the Journal of Molecular Biology. Also reproduced from the published paper (6), this method is similar to the one published in JBC but uses a modified information content function compared to a series of templates' frequency tables rather than just the number of mismatches for the criterion of the tests performed.

RESULTS AND DISCUSSION

Neural Network

Using test set A, it was found that trimming the weights to a total of 81 input units out of a possible 581 gave the best results for the 49 group training set. This significantly increased the ability of the network to generalize (Fig. 3). With the content units, trimming the input units improved the results with both test sets. However, when content units were not included, trimming actually worsened the results with test set B. This indicates that trimming input units is not always effective and that there is a difference between test set A and test set B (see below). Although it was not obvious if or by what amount the number of input units should be increased for the 98 group training set, we used 97 input units and one bias giving a total of 98 variable weights. In keeping with this we used 146 input units for the combined (training set B plus test set B) 147 groups of promoters. The final network's weights after 3528 rounds of training are shown in Table 2. As expected, most of the inputs units cluster around the conserved hexamers. It can also been seen that the requirement for high A+T content is much stronger in the −10 region than in the −35 region.

The final network was one of many that we experimented with using the older promoters. In particular, we found that hidden units did not improve accuracy. We also tried selecting the input units whose inputs were statistically the most significantly different from what would be expected at random instead of trimming the input units as described in Methods. This is similar to the approach taken by Abremski *et al.* (10), except that we considered individual bases and content units while Abremski *et al.* considered individual bases and base pairs. Trimming gave the best results however, possibly indicating that some non-linear information was learned. Many window lengths for the content input were also tried, most of whose accuracy with test set A could be improved by trimming. A window length of 12 gave the best results and was therefore used in the final network.

Comparison with Other Methods

Table 3 shows the results of comparing the predictive power of the different methods. Where possible, thresholds were adjusted for the minimum number of false positives with the number of false negatives below 20%. In this test, Mulligan *et al.*'s and Staden's methods performed better than the more sophisticated

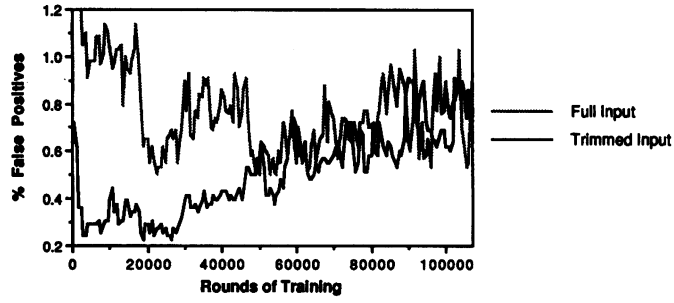

Figure 3. The prediction accuracies of two networks against test set A versus the number of rounds of training with training set A are shown. The curve marked full input is with the full 581 input units, while the curve marked trimmed input is with 81 input units.

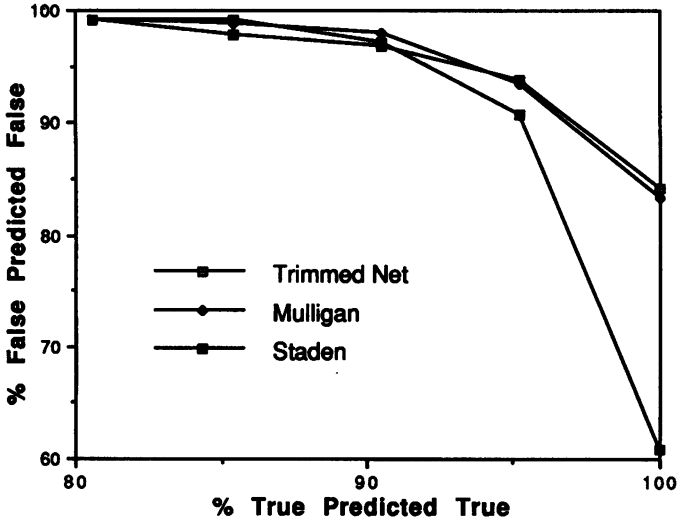

Figure 4. The accuracies of our neural net, Mulligan *et al.*'s method, and Staden's method at different levels of false negatives are shown. Training set B was used for training and test set B for evaluation of prediction accuracy.

models. Staden's method appears slightly better here but Mulligan *et al.*'s method performed better than Staden's when thresholds were set to lower levels of false negatives, despite the fact that Hawley and Reynolds (12) used Staden's log frequency criterion for their alignment (the alignment of Hawley and Reynolds was used for the test set). Attempts were made to further enhance these two methods by adjusting the training set, using small sample approximation in calculating base frequencies and adjusting the position and number of base frequencies used. This did not improve the prediction rate even when test set B was used to adjust those parameters (results not shown). The results of our network were found to be quite similar to Mulligan *et al.*'s method for all levels of false negatives (Fig. 4). Here our network fared slightly better than a neural net did in the analysis of ATP-binding motif prediction, where Hirst and Sternberg (19) found a statistical method to perform marginally better than a perceptron type network.

Of the more sophisticated methods, O'Neill's rule based methods appeared to be too specific to the training data. His

Table 2. Network Weights.

Bias Weight −048														
	a	c	g	t	a+c	a+g	a+t	a	c	g	t	a+c	a+g	a+t
t	−	−	−	−	−	−	−							
a	−	−	−230	−	−	−	−							
a	−	−	224	−	−	−	−							
c	−	−	−	−	−	−	−							
t	−240	−	−	−	−	−	−		Twelve Base Length Windows					
a	−	−	−	−	278	−	−	−	−	−358	−	−	−	−
a	−	−	−	−	292	−	−	−	−	−127	−	−	−	−
a	−	−332	−	−	−	−	162	−	−	−	−186	−	−	−287
t	−	−	−328	−	−	−	−	−	−	−123	−	−	−	−
a	−	−	−	−249	−	191	−	−	−300	−	−	−	−	−
a	−	−380	−	−	−	−	−	−	−	−	−	−	−	−
t	−	−	−150	−	−	−223	−	−	−371	−	−	−	−	−
t	−	−394	−	−	−	−	−	−	−	−	−	−	−	−
c	−	385	−	−	162	−	−404	−	−	−	−	−	−	−
T	−356	−219	−	743	−551	−595	−	−	−	−	−	−	−	−
T	−	−	−	483	−472	−335	229	−	−	−	−	−	−	−
G	−252	−	728	−296	−504	−	−524	−	−	−	−	−	−479	−
A	538	−	−	−283	555	218	−	−	−189	−	−	−	−	284
C	−	445	−209	−311	448	−	−308	−	−	−	−	−	−	179
A	340	−341	−	−	−	273	361							
t	−	−	−	−	−	−	−							
t	220	−	−	−	185	−	−							
t	−	−	−	199	−285	−	−							
t	−	−	−	−	−	−	−							
a	−	−	−	−	−	−	−							
			Spacing Class	(21)	(20)	(19)	(18)	(17)	(16)	(15)				
			Spacing Weight	−548	−440	−450	515	891	419	−434				
t	−	−	−	−	−	−	−783	−						
a	−	−	−239	−	339	−	−							
a	−	−	−	−	−	−	−263							
a	−	−	−	−	−	−	−							
t	−	−	−102	−	−	−	−		Twelve Base Length Windows					
t	−454	−	−	269	−268	−503	−	−	−	−221	−	−	−	−
a	−	−	−	−318	−	−	−	−	−	−149	−	−	−	155
t	−	−	−	−	−354	−	−	−	−	−	−	−	−	234
g	−	−	211	−	−	−	−237	−	−108	−090	−	−	−	−
t	−	−	384	−	−	−	−476	−	−	−172	445	−	−	269
T	−452	−	−322	666	−416	−750	−	−	−	−196	395	−	−	−
A	1071	−304	−348	−514	791	747	581	−	−259	−	−	−	−	−
T	−	−	−	277	−161	−146	−	−	−378	−	−	−104	−	−
A	521	−242	−	−	303	522	−	−	−256	−	−	−	−	−
A	−	−	−	−397	227	338	−	−	−228	−158	−	−	−	−
T	−587	−323	−	1048	−887	−797	485	−	−163	−203	−	−	−	−
t	−	−	−	−	−	277	−	382	−	−217	−	313	−	−
a	−	−	−	−	−	−	−429	−	−	−132	−	−	−	−
a	−	−	−	−	−	−	−	−	−	−163	−	−	−	−
a	−	−	−	−251	−	−	−	−	−	−255	−	−	−	−
c	−	−	−	−	−229	−	−328	−	−295	−	−	−	−	394
c	−	−	−	−	−	−	−508							
a	−	−	−399	−	−	−	−							
a	−	−371	−	−	−239	−	−							
t	−	−	−	−	−	−	−							
t	−	−	−234	−	−	−	−							
g	−	−	−	−	−	−	−							

Network weights are shown multiplied by 1000.

method does not allow for adjustment with a simple parameter, but when the threshold of our network was adjusted to the high false negative level of 71.4% (28.6% of promoters correctly predicted) it produced only 0.02% false positives compared to O'Neill's 0.04% and 0.05%. It should be noted, however, that those rules are only for 16,17 and 18 base spacing class promoters and that using our general promoter database for evaluation probably lowered their prediction accuracy. Alexandrov's weight matrices also did not perform well on our test set.

Comparison with Other Neural Nets

O'Neill (9) and Demeler and Zhou (11) have previously reported excellent results with neural networks. When our network was trained and tested with their data sets however, it was found to give comparable results (Table 4). The difference in prediction rates with these different data sets seems to be explainable to a large extent by the differences in information content of the combined training and test sets (Fig. 5). It should be noted that

Table 3. Comparison of the Prediction Accuracy of Different Methods.

Method	Threshold	Training Promoter %Correct	Test Promoter %Correct	Test Non-Promoter %False Positives
Mulligan	42.0	85.2	80.6	0.85
Staden	−76.8[a]	84.7	80.3	0.83
	−59.6[b]	89.8	81.6	0.84
O'Neill JBC	6 tests	57.7[c]	24.5	0.04
O'Neill JMB	6 tests	65.3[c]	27.6	0.05
Alexandrov[d]	12.9	85.2	82.7	3.40
Alexandrov[e]	37.7	89.3	81.6	2.56
Our Neural Net	0.604	100	80.6	0.86

a) sum of three regions plus gap penalty, b) +1 region excluded, c) includes some sequences outside the training set used, d) published matrix, e) matrix obtained through personal communication.

Table 4. Prediction Results with Different Data Sets.[a]

Data Set	Our Neural Net		Mulligan		Reported Results	
	+	−	+	−	+	−
O'Neill	81	0.03	81	0.06	~80	<0.1
	100	0.23	100	0.77		
Demeler	83	0.93	83	0.28		
	100	1.58	100	5.52	100	1.6
Harley	80.6	0.86	80.6	0.85		
	100	15.85	100	16.63		

a) The + column contains the percent of promoters correctly predicted while the − column contains the percent of non-promoters mistakenly predicted to be promoters.

both of their data sets are essentially subsets of our data set built from all of Harley and Reynolds' (12) compilation and we therefore consider them to be less general. In particular Demeler and Zhou's data set contains only promoters that were identified during or before 1983 and O'Neill's data set contains only promoters of the 17 base spacing class.

Extended Data Sets

As can be seen in Table 3, while showing a comparable performance to Mulligan *et al.*'s method on the testing set, our neural network did much better than Mulligan *et al.*'s method on the training set. When the threshhold was set for 80% correct prediction of test promoters our neural network correctly predicted all of the training promoters. This result suggested that a neural network might do better if either the training data was expanded or if the restrictions limiting homology between the data sets were relaxed. We tried both, separately and together.

First we tried enlarging the data set by training the network on all of training set B plus 25 of the 49 test set B groups and testing with the 24 remaining test set B groups. As before the number of input units was trimmed to equal 1 less than the number of homology groups in the training data. The number of rounds of training was kept as close as possible to that determined before with test set A, with the restriction that the number of rounds of training was always set to be a multiple of the size of the training set (see Methods section Database). The 25 group and 24 group halves were then interchanged and the results were averaged. The results were again comparable to Mulligan *et al.*'s method when the respective training data was used to recalculate Mulligan *et al.*'s weights (results not shown). Since this did not increase the neural network's relative performance we tried loosening the restriction on homology (see Methods section Database) to merely forbidding identical sequences from being in both training and testing sets. With this looser restriction 100 sequences, corresponding to 83 homology groups, were found in Harley and Reynolds' compilation that were not present in Hawley and McClure's compilation. Using these 83 groups as an extended test set B, our network, with its original training set B, and Mulligan *et al.*'s method, with its original weights, again showed comparable performance (results not shown). When the new, less restricted extended test set B was used to enlarge the training data however, the results were different. Here we divided the 100 sequences making up the extended test set B randomly into two halves of 50 sequences each. Upon grouping, these two halves gave 46 and 49 homology groups respectively. The training data was enlarged by including one half of the extended test set B with the training data and using the other half for testing. Added to the 98 groups from Hawley and McClure's compilation (training set B) this gave training sets of 144 and 147 groups respectively. Mulligan's weights were also recalculated using the base frequencies of the new training sets. Fig. 6(a) shows the results from predicting half of the extended test set B with our network trained with training set B plus the other half of the extended test set B. Fig. 6(b) shows the results obtained when the roles of the 'halves' above were interchanged, i.e. when the extended test set B groups used for evaluation in Fig 6(a) were added to training set B and the extended test set B groups added to the training set in Fig. 6(a) were used for evaluation. As can be seen in Fig. 6, the resultant neural networks clearly performed better than Mulligan *et al.*'s method when false negatives were kept to within 5% or less.

New Promoters

Interestingly, while a large database is clearly favorable to prediction schemes, especially neural networks, the question of whether simply identifying more promoters will easily allow for

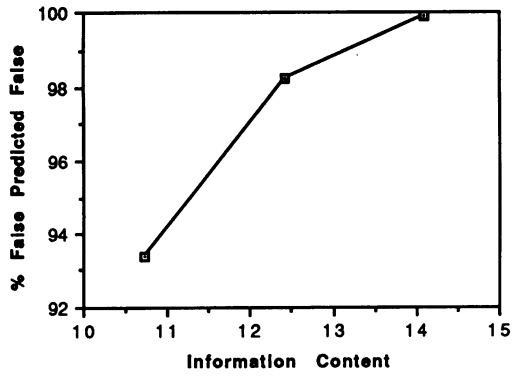

Figure 5. The correlation between the information content of three data sets (our test and training set B, O'Neill's data set and Demeler's data set) and the prediction accuracies obtained with the respective data sets is shown. The information content (in bits) is the sum of 52 base positions. The prediction accuracies are those attained by our network with the respective data sets when the network's threshold was adjusted to as close as possible to, without exceeding, a 5% level of false negatives.

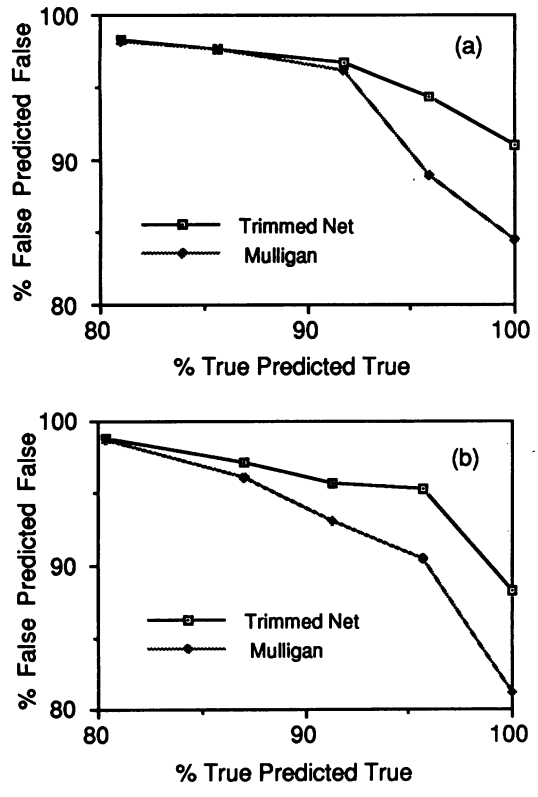

Figure 6. The accuracies of our neural net and Mulligan *et al.*'s method at different levels of false negatives are shown. The results with (a) the 144 group extended training set and (b) the 147 group extended training set.

better prediction of unknown promoters or not is a difficult one. This is because there is no guarantee that the characteristics of new promoters will be exactly the same as those of known promoters. As a comparison of the percent of false positives in Fig. 3 and Table 3 indicates, test set B, which consists mainly of promoters identified between 1983 and 1987, was consistently harder to predict than test set A, which consists of promoters identified in or before 1983. The reason for this is not obvious from the average number of matches to the hexamers or the average A+T content of the earlier and later promoters (8.2 matches vs. 8.0 matches and 62% vs. 57% A+T content). The information content between the two groups does differ somewhat however, with the older promoters having 11.20 bits of information in their most conserved 30 positions and the newer promoters having only 9.94 bits of information in their 30 most conserved positions. This is despite the smaller size of the newer group, which would tend to give it a higher amount of information due to random noise. It may be that later experimental techniques allowed for the identification of weaker promoters, but we have not compared published strengths of promoters to confirm this.

It should also be mentioned that recent findings involving synthetic promoters have challenged the basic framework upon which the prediction methods compared here are based (20–24). In particular the positioning of the hexamers relative to the transcription start site and the spacing between the two hexamers appears not to be as constrained as previously thought. Although it has not been shown that such 'unusual' promoter structures are common in natural promoters their existence should be considered in future prediction schemes.

CONCLUDING REMARKS

Roughly speaking, if the number of known sequences is less than the number of features needed to describe their common functional motif, a neural network seems unlikely to produce good results. In this case, a weight matrix produced by the extremely simple operation of summing the base frequencies after alignment may, as it did here, produce quite reasonable results. When, as is the case of *E.coli* promoters, there is a sizable but limited database the form and number of inputs shown to a neural net has been shown to be important. Selecting the correct input has, in fact, been a focal point not only of this work, but also of previous researchers (8,10).

In principle, the use of hidden units in neural nets makes it possible to learn high order correlations between sequences and function. In this work however, we were not able to obtain better results by using hidden units. This finding is similar to the reported results of using neural networks for predicting protein secondary structure (16,17,25,26). Moreover, Demeler and Zhou (11) reported that the number of hidden layer units does not have a significant effect on prediction accuracy. O'Neill (9) reported good results predicting promoters using hidden units but did not compare his results with a perceptron architecture network.

In this discussion we should mention some potential problems with our choice of using coding regions at random for our negative data. One potential problem is that, as promoters are sometimes located in the coding regions of other genes, it is possible that our negative data contained some (albeit a very small percentage) actual promoters. Neural networks are known for being robust against low levels of noise and we felt that a possible very low level of contamination of the negative data with promoter sequences could be tolerated. Perhaps a more serious possibility was that our network might memorize the characteristics of coding regions and therefore have a poor ability to reconize non-promoter sequences from non-coding regions. It seemed unlikely to us that the network could learn the characteristics of coding regions as the coding regions were presented randomly with respect to frame and strand, and the perceptron archeticture would not allow for learning three base

periodicities. Moreover, the base composition of the coding regions chosen did not show any strong biases with a 27% thymine composition being the largest deviation. Interestingly however, when the methods listed in Table 3. were tested with 50% A+T content random data for negative test data, all of the methods prediction accuracies dropped slightly. Mulligan's method for example fell from 0.85% false positives to 1.42% false positives. Our neural network's accuracy fell slightly more from 0.86% false positives to 1.89% false positives. These differences are small however, and shouldn't change the validity or usefulness of the results reported in this paper. In fact, when comparing our network with the results of O'Neill's and Demeler *et al.*'s networks we trained our network with 50% and 60% A+T content random data respectively. The results in (Table 4) indicate that our network's success did not rely upon using coding regions for negative data and also show that it was not necessary to use promoter down mutations as O'Neill did to achieve his reported accuracy.

In conclusion, we have improved the generalization ability of a neural network for predicting promoters by trimming input units with a new technique that is applicable to any neural network used in sequence analysis. The resulting trimmed network was found to be superior to the best alternative method on training data and roughly equal to or better than it on testing data. In particular, the neural network achieved good results when the data set was large and the level of false negatives was kept low.

ACKNOWLEDGEMENTS

P.B.H. was supported by Monbusho Scholarship from the Ministry of Education, Science and Culture of Japan. This work was partly supported by a grant from the Human Frontier Science Program.

REFERENCES

1. Harr, R.,Häggstrom M., and Gustafsson P. (1983) *Nucl. Acids Res.* **11**, 2943–2957.
2. Mulligan, M.E., Hawley D.K., Entriken R., and McClure, W.R. (1984) *Nucl. Acids Res.* **12**, 789–800.
3. Staden, R. (1984) *Nucl. Acids Res.* **12**, 505–519.
4. Nakata,K., Kanehisa,M. and Maizel,J. (1988) *CABIOS* **4**, 367–371.
5. O'Neill, M.C. (1989) *J. Biol. Chem.* **264**, 5522–5530.
6. O'Neill, M.C. (1989) *J. Mol. Biol.* **207**, 301–310.
7. Alexandrov, N.N. and Mironov, A.A. (1990) *Nucl. Acids Res.* **18**, 1847–1852.
8. Lukashin, A.V.,Anshelevich, V.V., Amirikyan, B.R., Gragerov, A.I.and Frank-Kamenetskii, M.D. (1989). *J. Biomolec. Structure and Dynamics* **6**, 1123–1133.
9. O'Neill, M.C. (1991) *Nucl. Acids Res.* **19**, 313–318.
10. Abremski, K., Sirotkin, K., and Lapedes, A. (1991) *Los Alamos National Laboratory Technical Report* LA-UR-91–729.
11. Demeler, B. and Zhou, G. (1991) *Nucl. acids Res.* **19**, 1593–1599.
12. Harley, C.B. and Reynolds, R.P. (1987) *Nucl. Acids Res.* **15**, 2343–2361.
13. Hawley, D.K. and McClure, W.R. (1983) *Nucl. Acids Res.* **11**, 2237–2255.
14. Schneider, T. D., Stormo, G. D., Gold, L. and Ehrenfeucht, A. (1986) *J. Mol. Biol.* **186**, 117–128.
15. Rumelhart, D.E., Hinton, G.E., and Williams, R.J. in 'Parallel Distributed Processing: Explorations in the Microstructures of Cognition.' pp. 318–362, Cambridge, Mass. MIT Press, 1986.
16. Qian, N. and Sejnowski, T. (1988) *J. Mol Biol.* **202**, 865–884.
17. Kneller, D., Cohen, F. and Langridge, R. (1990) *J. Mol Biol.* **214**, 171–182.
18. Gentz, R. and Bujard, H. (1985) *J. Bacteriol.* **164**, 70–77.
19. Hirst, J.D. and Sternberg,M.J.E. (1991) *Protein Engineering* 4 615–623
20. Horwitz, M.S. and Loeb, L.A. (1988) *J. Biol. Chem.* **263**, 14724–14731.
21. Jacquet, M-A., Ehrlich R., and Reiss, C. (1989) *Nucl. Acids Res.* **17**, 2933–2945.
22. Collis, C.M., Molloy, P.L., Both, G.W., and Drew, H.R. (1989) *Nucl. Acids Res.* **17**, 9447–9468.
23. Jacquet M.-A., and Reiss C. (1990) *Nucl. Acids Res.* **18**, 1137–1143.
24. Koroleva, O.N. and Drutsa, V.L. (1991) *FEBS Letters.* **278**, 207–210.
25. Holley, L. and Karplus, M. (1989) *Proc. Natl. Acad. Sci. USA* **86**, 152–156.
26. Storlorz, P., Lapedes, A. and Xia, Y. (1991) *Los Alamos Laboratory Technical Report* MS B213