Back-Propagation Learning on Ribosomal Binding sites in DNA sequences using preprocessed features

Submitted to: World Congress on Computational Intelligence
IEEE International Conference on Neural Networks

Lorien Y. Pratt Lauren L. Tracy
Dept. of Mathematical and Computer Sciences
Colorado School of Mines
Golden, CO 80401

Michiel Noordewier
Dept. of Computer Science
Rutgers University
New Brunswick, NJ 08903

Abstract

Several studies have explored how neural networks can be used to find genes within regions of previously uncharacterized deoxyribonucleic acid (DNA). This paper describes the creation of a neural network training set for determining which part of a DNA strand codes for an important genetic feature called a **Ribosomal Binding Site**, or **RBS**. Based on previous research on detecting other genetic features, this data set contains preprocessed features that reflect biologically meaningful patterns in the raw base pair [ACTG]* language. We also describe preliminary empirical results indicating neural network performance that is superior to all other automated methods for detecting RBS's.

1 Introduction

NEURAL networks have recently been used for a variety of applications [Maren *et al.*, 1990]. Comparisons between neural networks and competing techniques have also shown that neural network performance is competitive to more traditional methods on many tasks [Shavlik *et al.*, 1991, Weiss and Kulikowski, 1991]. Probably the most successful paradigm in which neural networks have been applied is called **classifier learning**. In this approach, a **classification algorithm** is automatically constructed, given as input a set of training examples that illustrate the desired input and output of the resulting classifier.

This paper describes how classifier learning can be applied to the detection of an important region on a deoxyribonucleic acid (DNA) molecule. Our general approach is to preprocess raw DNA data, which is over the base pair alphabet [ACTG]*, into higher-level features that are more biologically meaningful. This approach is based on that described by Hirsh and Noordewier, 1994, who used a superset of our features to improve the performance of classification networks that detected two other sorts of important DNA patterns, called **splice junctions** and **promoters**.

We begin, in Section 2, by giving some biological background to the importance and the state of the art in RBS detection. We then describe, in Section 3, how our data set was created. Finally, Section 4 describes the results of preliminary experiments to analyze the ability of networks trained using this data set to accurately and reliably locate RBS's.

2 Biology background

DNA is a molecule in the nucleus of a cell that contains a code for the creation of proteins, which are the building blocks of organisms. For example, red blood cells contain the protein **hemoglobin**, and hair contains **keratin**. The DNA code is over the alphabet: [ACTG]*, where A (*adenine*), C (*cytosine*), T (*thymine*), and G (*guanine*) are called **bases**. All of the information describing an organism is coded by sequences of these letters, which correspond to parts of the DNA molecule. The goal of Human Genome Project [Lander *et al.*, 1991] is to "decode" DNA – to determine the exact meaning of each position on the DNA strand. Important recent breakthroughs in this effort include the discovery of a region of DNA that encodes for breast cancer and another that codes for Alzheimer's disease [Cowley, 1993].

One approach to decoding DNA is to start by determining the location of functional units along the strand called **genes**. There are two general approaches to finding genes. First, laboratory experiments can be performed to manipulate the DNA strand in such a way as to pinpoint gene positions. Secondly, the results of those experiments can be used to determine how, simply by examining the base pair sequence, genes can be detected.

Our research concerns this latter approach. It is strictly computational – we start with information about gene positions gathered in the laboratory, and finish with a system that is able to take a base pair sequence as input, and determine whether a region that

indicates a meaningful sequence indicating a gene is contained within it. Since human DNA contains roughly three billion base pairs, and since it is a much simpler laboratory process to list DNA base pairs than to find genes, this approach can have a substantial impact on genetic research.

2.1 The ribosomal binding site detection task

The process of converting DNA to proteins is called **protein synthesis**. Roughly speaking, this process is as follows:

1. An RNA copy is formed from the DNA template (this is called *transcription*).
2. This RNA copy is converted to **messenger RNA (mRNA)**.
3. Protein complexes, known as *ribosomes*, bind to the mRNA strand.
4. The ribosome/mRNA complex forms proteins.

There is a wide body of literature on laboratory and computational methods to analyze this process. Previous computational efforts involving neural networks include:

- [Towell and Shavlik, 1992], who showed that a neural network that is initialized with knowledge about genetics is more accurately able to detect a **promoter**, which labels the start of transcription. This method outperformed other computational methods, including an expert system.
- [Stormo *et al.*, 1982], who showed that a neural network can be used to detect **ribosomal binding sites**, which are the DNA encodings for positions on the mRNA strand to which ribosomes will bind during step 3 of the above process.
- [Hirsh and Noordewier, 1994], who constructed neural network and decision tree classifier learners to detect both promoters and also **splice junctions**, which indicate sequences in DNA that are excised before RNA translation begins. They showed that, by converting $[ACTG]^\star$ strings into more meaningful biological **features**, classifier performance could be improved.

This paper applies the feature preprocessing method of Hirsh and Noordewier to the ribosomal binding site detection task studied by Stormo *et al.*. We describe the construction of a data set and a neural network that examines a number of DNA base pairs from the organism *Escherichia coli (E. Coli)* and determines if an RBS is present.

3 Approach

Our approach consisted of four steps. We first extracted base pair sequences known to contain RBS's from a genome database file. These sequences were aligned so that the suspected RBS start position was in its center. We also extracted sequences that were not labelled as containing an RBS (negative examples). We then converted both the positive and negative sequences to high-level features, following the approach of Hirsh and Noordewier. Finally, we performed a preliminary experiment to train a neural network to recognize RBS sites using this data set. We obtained a somewhat reliable score of 62.4% correct using a single train/test set split.

These steps are described in more detail in the following sections.

3.1 RBS extraction and alignment

[Stormo *et al.*, 1982] describes experiments to determine the optimal number of base pairs to examine, or **window size**, when trying to detect RBS's. It's important that the window size be large enough to contain enough contextual information so that RBS's can be detected, but small enough that network training time is reasonable, and good generalization is possible because the number of network parameters (weights) is kept to a minimum.

We chose to use a window containing 50 base pairs. We began by extracting a window that was 70 base pairs wide from the Genbank version 74.0 bacterial sequences file: gbbct.seq. We restricted extraction of these windows to sequence file entries labelled "Complete CDS", indicating that the complete coding sequence was listed. We extracted base pair strings that had 34 bases to the left of a position indicated in the sequence header to begin an RBS region, and 35 bases to its right. Strings without enough bases in the sequence entry to fill this size-70 window were discarded.

Since GenBank entries are known to not necessarily be uniform in RBS site location, we then **aligned** our sequences by shifting them left or right by at most 5 base pairs. Our alignment procedure compared each new sequence with all those previously examined, and chose the alignment position for the new sequence which yielded the highest proportion of identical base pairs in the same position to previous strings, among the eleven allowable shift positions (5 left, 5 right, and one for no shift). After this shift was complete, the strings were truncated to a window of 50 base pairs wide.

3.2 Extraction of negative training data

For every positive training example created as described above, we also created a negative example. This was done by extracting a 50-base pair string

from the same coding sequence that the positive example was extracted from, except that the base pair string ended five base pairs before the indicated start of the RBS.

3.3 Conversion to high-level features

After creating both positive and negative training examples over the [ACTG]* language, we converted them to higher level features, closely following the successful approach described by Hirsh and Noordewier.

Our higher-level features were a subset of those used by Hirsh and Noordewier that we judged were relevant to RBS detection. As with their features, ours fell into three general classes. The first, **site-specific** features, encoded simple patterns which were found empirically to be relevant to DNA. For each pattern, a single boolean higher-level feature was set, which represented the presence (1) or absence (0) of the pattern. The second class, called **conformational features**, encoded physical and chemical properties of a sequence. These more complex patterns were also encoded as a boolean higher-level feature. Finally, a number of features were used which measured the ratio of different base pairs to others. These were converted to a number in [0,1].

We now describe each feature used in turn.

3.3.1 Helical Parameters

Helical parameters represent aspects of the shape of the double helix that are believed to be relevant for RBS creation during transcription. These features were detected using simple rules, shown in Figure 1. In this table, "r" stands for G or A (representing the class of nucleotides known as purines), and "y" stands for T or C (representing the class of nucleotides known as pyrimidines). In all, there were eighteen helical parameters. Note that there is some overlap, so that, for example, any sequence containing a twist2a will also by definition contain a twist1a feature.

twist1a	→	r,r,r,r,y.	twist1b	→	y,r,y,y,y.
twist2a	→	r,r,r,r,y,r.	twist2b	→	y,r,y,y,y,y.
twist3a	→	r,r,r,y,r.	twist3b	→	r,y,y,y,y.
roll4a	→	r,r,r,y,y.	roll4b	→	r,y,y,y,r.
roll5a	→	r,r,y,y,y.	roll5b	→	y,r,r,r,y.
roll6a	→	r,r,y,y,y,r.	roll6b	→	y,r,r,r,y,y.
twist7a	→	r,y,r,r,r.	twist7b	→	y,y,r,y,r.
twist8a	→	y,r,y,r,r.	twist8b	→	y,y,y,r,y.
twist9a	→	y,r,y,r,r,r.	twist9b	→	y,y,y,r,y,r.

Figure 1: Helical parameter higher-level features that were extracted.

3.3.2 Site-Specific information

Hirsh and Noordewier also used short nucleotide sequences called **motifs** as higher-level features. These three features are shown in Figure 2. In the gtg_pair feature, the ellipse indicates any number of base pairs within the window, surrounded by gtg subsequences.

gtg	→	"gtg"
gtg	→	"cac"
gtg_pair	→	gtg, ..., gtg.

Figure 2: Site-specific "motif" features.

3.3.3 Ratio-content

For each of the six different base pair pairings: AT, CT, GT, CA, CG, GA, we measured the relative proportions of the two base pairs within the window. This produced six new features, all with values in [0,1].

3.4 Neural network training

Following the above procedure on the gbbct.seq file produced 1970 examples. These were split into a single train and test set, with 1000 training examples and 970 test examples.

To determine a good hidden unit count, we chose to follow [Shavlik *et al.*, 1991] and to use a number that was approximately 10% of the total number of input and output units. This resulted in three hidden units and an overall network topology of 29-3-1.

To determine the learning rate (η) and momentum (α) parameters to back-propagation, we ran networks on the training data with 17 different pairs of these values to 10,000 epochs each. The 17 different (η, α) pairs were chosen based on a local search through a space of these values, starting at the four extremal points of (0.1,0.9), (0.1, 0.1), (0.9, 0.1), and (0.9, 0.9). The best final TSS of 202.2 was obtained with $\eta = 0.1$, $\alpha = 0.25$.

4 Results

We considered the learned network to correctly classify a test data item when it produced an output activation > 0.5 for a target activation of 1 and < 0.5 for a target activation of 0. By this measurement, the network that was trained for 10,000 epochs with parameters as described above achieved $605/970 = 62.4\%$ correct on the testing data.

5 Conclusion

This paper has described how a data set with preprocessed features was generated and used to train a neural network on the problem of ribosomal binding site detection in *E. coli*. The most recent work of which we are aware that addressed the detection of RBS's based on nucleotide sequences is that of [Stormo *et al.*, 1982]. That paper focused on determining the best window size in which to do RBS

detection. Therefore its empirical results were inconclusive, since their network was evaluated on only 10 genes. Their best performance was with a network that had a window size of 101 base pairs – out of 10 possible genes it found 7, but labelled 5 nongenes as genes also.

We have presented results of a more rigorous empirical evaluation of a neural network for this task, and achieved better results. We view this work as a preliminary study, because we see that there is substantial room for improvement over the 62.4% level. There are several more features that could be added to the network, which may improve performance. Also, if network training were biased so as to eliminate false negatives, then a performance of 62.4% would be adequate to allow this network to be used as a data filter. The network could be used to eliminate sequences that it was certain did not contain RBS sequences, and to pass along the remainder to other more traditional methods, thereby reducing the amount of data that needed to be examined.

The training set could also be improved by extracting negative training examples less systematically. It's possible that, by always extracting negative examples from right before the RBS, we are biasing the learning process because there may be a systematic pattern in this data as well.

A third important extension to this work is more thorough empirical testing. Although 940 examples is a larger set than has previously been used to evaluate a network on this task, this is not a large enough number to give a satisfyingly reliable estimate of future performance. Future work will include gathering more data on this task, and also doing cross-validation evaluation [Weiss and Kulikowski, 1991].

Our final direction for future work is to use the technique of **transfer** between neural networks reported in [Pratt, 1993] to use the network constructed for *E. coli* to facilitate the training of a network for a different organism.

6 Acknowledgements

Maxime Pitard wrote many of the Genbank preprocessing code that we used; this project would not have been possible without his help.

References

[Cowley, 1993] Geoffrey Cowley. Family matters. *Newsweek*, 72(23):46–52, December 6 1993.

[Hirsh and Noordewier, 1994] Haym Hirsh and Michiel Noordewier. Using background knowledge to improve inductive learning of dna sequences. In *To appear: Proceedings of the Tenth IEEE Conference on Artificial Intelligence for Applications, San Antonio, Texas*, March 1994.

[Lander *et al.*, 1991] E. S. Lander, R. Langridge, and D. M. Saccocio. Computing in molecular biology: Mapping and interpreting biological information. *IEEE Computer*, pages 6–13, November 1991.

[Maren *et al.*, 1990] Allianna J. Maren, Craig T. Harston, and Robert M. Pap. *Handbook of neural computing applications*. Academic Press, 1990.

[Pratt, 1993] L. Y. Pratt. Discriminability-based transfer between neural networks. In C.L. Giles, S. J. Hanson, and J. D. Cowan, editors, *Advances in Neural Information Processing Systems 5*, pages 204–211. Morgan Kaufmann Publishers, San Mateo, CA, 1993. Also available via anonymous ftp to franklinite.mines.colorado.edu: pub/pratt-papers/pratt-nips5.ps.Z.

[Shavlik *et al.*, 1991] J. W. Shavlik, R. J. Mooney, and G. G. Towell. Symbolic and neural net learning algorithms: An experimental comparison. *Machine Learning*, 6(2):111–143, 1991.

[Stormo *et al.*, 1982] Gary D. Stormo, Thomas D. Schneider, Larry Gold, and Andrzej Ehrenfeucht. Use of the 'perceptron' algorithm to distinguish translational initiation sites in *e. coli*. *Nucleic Acids Research*, 10:2997–3011, 1982.

[Towell and Shavlik, 1992] Geoffrey G. Towell and Jude W. Shavlik. Interpretation of artificial neural networks: Mapping knowledge-based neural networks into rules. In *Advances in Neural Information Processing Systems 4*, pages 977–984, San Mateo, CA, 1992. Morgan Kaufmann.

[Weiss and Kulikowski, 1991] Sholom M. Weiss and Casimir A. Kulikowski. *Computer Systems that Learn*. Morgan Kaufmann, 1991.