BIOINFORMATICS

Vol. 15 no. 9 1999
Pages 723–728

Neural network input representations that produce accurate consensus sequences from DNA fragment assemblies

C. F. Allex[1,2], J. W. Shavlik[1] and F. R. Blattner[2,3]

[1]Computer Sciences Department, University of Wisconsin – Madison, 1210 West Dayton Street, Madison, WI 53706, [2]DNASTAR Inc., 1228 South Park Street, Madison, WI 53715 and [3]Genetics Department, University of Wisconsin – Madison, 445 Henry Mall, Madison, WI 53706, USA

Received on October 12, 1998; revised on April 2, 1999; accepted on April 23, 1999

Abstract

Motivation: *Given inputs extracted from an aligned column of DNA bases and the underlying Perkin Elmer Applied Biosystems (ABI) fluorescent traces, our goal is to train a neural network to determine correctly the consensus base for the column. Choosing an appropriate network input representation is critical to success in this task. We empirically compare five representations; one uses only base calls and the others include trace information.*

Results: *We attained the most accurate results from networks that incorporate trace information into their input representations. Based on estimates derived from using 10-fold cross-validation, the best network topology produces consensus accuracies ranging from 99.26% to >99.98% for coverages from two to six aligned sequences. With a coverage of six, it makes only three errors in 20 000 consensus calls. In contrast, the network that only uses base calls in its input representation has over double that error rate: eight errors in 20 000 consensus calls.*

Contact: *allex@cs.wisc.edu*

Introduction

We have applied neural networks to the task of determining the consensus base in a column of aligned DNA sequences. The problem we addressed is referred to as *consensus calling* and is briefly described in Figure 1.

Accuracy in consensus sequences is an important concern; the National Human Genome Research Institute (NHGRI) set a standard for sequencing accuracy at 99.99% (NHGRI, 1998). Unfortunately, the error rate for sequences in GenBank has been estimated to be from 0.3 to 0.03% (Lawrence and Solovyev, 1994)—much higher than the standard. When imperfect DNA sequences are translated, the effect on the resulting protein sequence can be substantial. Even the mutation of a single amino acid can cause critical changes in the character of a predicted protein. Furthermore, the deletion or insertion of bases can result in frame shifts that lead to dramatically increased error rates and the failure to recognize open reading frames when the DNA is translated.

Currently, sequencing accuracy is significantly dependent upon careful human examination and editing of consensus sequences in fragment assemblies. The hand process is time consuming, expensive and error prone, making it unsuitable for large-scale sequencing projects. Automatic methods such as ours, that produce highly accurate consensus calls, reduce errors and alleviate the need for human editing.

One significant way that our system for consensus calling differs from most existing methods is that it directly processes information on the shape and intensity of Perkin Elmer Applied Biosystems (ABI) fluorescent traces. Other methods, such as those in the *TIGR Assembler* (Sutton *et al*., 1995), and the *Staden Package* (Bonfield *et al*., 1995), examine only previously determined base calls when calculating the consensus.

Two existing assemblers that do consider trace characteristics are *Phrap* (Green, 1997) and DNASTAR's *SeqMan II*. To make a consensus call, *Phrap* chooses the base call in an aligned column with the highest quality trace as determined by its companion base-calling program, *Phred* (Ewing and Green, 1988; Ewing *et al*., 1998). In *SeqMan II*, the consensus is determined by a method we developed during earlier work (Allex *et al*., 1997). The method extracts and sums information about the shape and intensity of the traces in an alignment. The sums are used as evidence in determining the most likely consensus call.

Another difference between our system and others is our use of neural networks. Figure 2 contains a brief description of the operation of neural networks; details can be found in McClelland and Rumelhart (1986). Neural networks can be a powerful data analysis tool for problems in molecular biology (Baldi and Brunak, 1998). Their strength is in their ability to learn and use complex patterns such as those found in these types of problems. Despite this, the use of neural networks for tasks in DNA sequencing has been scarcely explored. In one promising example, neural networks are used

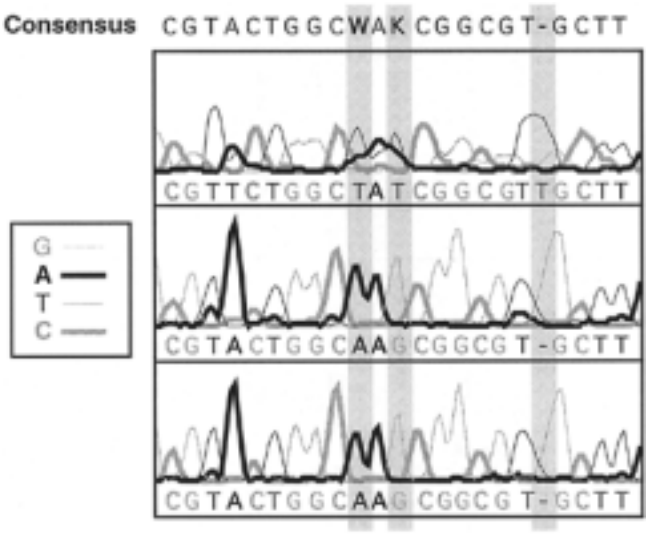
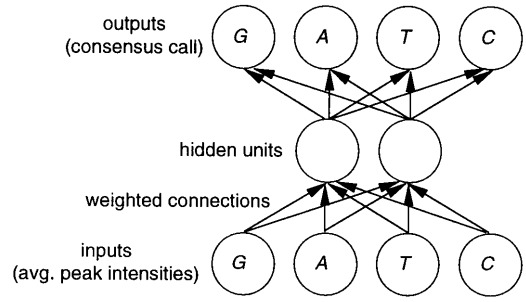


Fig. 1. Consensus calling. State-of-the-art sequencers such as the Perkin Elmer Applied Biosystems (ABI) 377 use fluorescent dye labeling to determine DNA fragment sequences (Ansorge *et al*., 1986; Smith *et al*., 1986). For each fragment, the sequencing process produces dye intensities in four sets of fluorescent traces. Here we have an example of three fragments that have been sequenced and aligned. For each fragment, traces and corresponding base calls output by ABI software are shown. Once sequences have been aligned, the consensus sequence, as listed above the alignment, is calculated. In most columns in this example, the base calls indicated by the traces exhibit total agreement. However, in the first two highlighted columns, the base calls and traces conflict and the appropriate ambiguity code is listed as the consensus call. (*W* indicates *A* or *T* and *K* indicates *T* or *G*.) In the rightmost highlighted column, a base call has been erroneously inserted in the first fragment and the consensus shows a gap, meaning no base exists there.



Inputs: Average relative *G*, *A*, *T*, and *C* trace peak intensities

Outputs: A consensus call for the aligned column

Categorized Examples

	Inputs				Desired Outputs			
example 1:	.32	.01	0	.03	1	0	0	0
example 2:	.05	0	.01	.35	0	0	0	1
...								
example *n*:	.38	.01	.04	0	1	0	0	0

Fig. 2. Neural networks. A feed-forward back propagation neural network learns to categorize patterns of inputs. Inputs are numerical representations of features of a problem. Typically, there is one output for each category of the problem; the desired output is 1 for the correct category and 0 otherwise. First the network is trained by processing a set of categorized examples (a training set). A categorized example is an instance of the problem that includes its inputs and desired outputs. During training, weighted connections in the network are adjusted so that the error in the actual output is reduced. Hidden units in the network aid by allowing the input representation to be transformed. When the difference between the desired and actual inputs is sufficiently low, training is halted and the network can be used to categorize previously unseen instances of the problem. Future accuracy of the trained network is estimated by measuring a trained network's performance on a disjoint set of testing examples. In this figure, we have an example of a simple neural network whose function is to call the consensus for a single aligned column of DNA bases when given inputs extracted from fluorescent traces. The network is given four inputs (the relative *G*, *A*, *T* and *C* trace intensity averages) and outputs a consensus call (*G*, *A*, *T* or *C*).

to make base calls in individual DNA sequences (Golden *et al*., 1993). Note that Golden's work calls bases in single sequences, whereas the work we describe determines the consensus for multiple aligned sequences.

System and methods

The ability of a neural network to categorize instances of a problem correctly is critically dependent upon the input representation (Baldi and Brunak, 1998). For our work, this problem can be expressed as follows.

Given: An aligned column of base calls and traces

Do: Represent the column as numerical inputs

We define four features of an aligned column that can be used singly or in combination to form input representations for a neural network. Two of the features use information extracted from fluorescent traces. We believe that much valuable information is lost when the traces are reduced to base calls. Our hypothesis is that a neural network can exploit the trace information to make consensus calls that are more accurate than those made with networks that use only base calls as inputs.

The inputs that use trace information are weighted by the quality of the trace so that more emphasis is given to better data. A description of the calculation of the quality values we use appears in Allex *et al*. (1997). One of the input features that uses fluorescent trace information captures the shape of

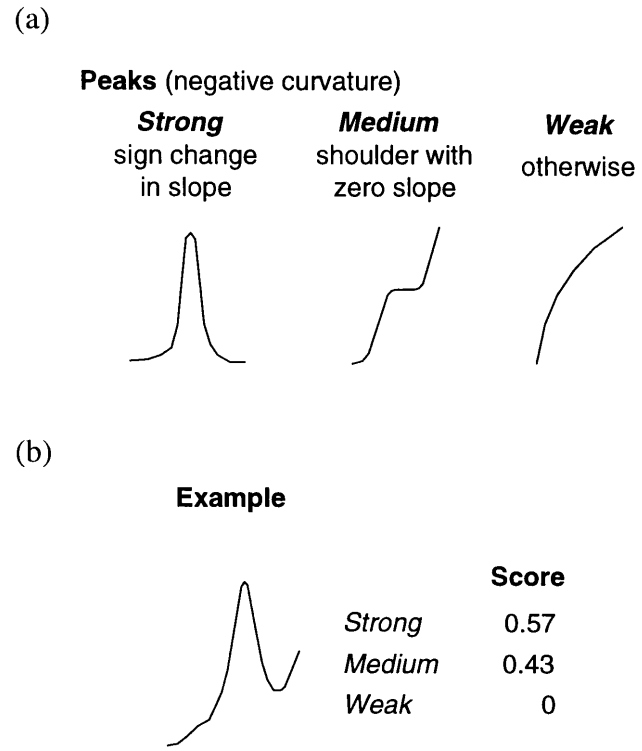


Fig. 3. *Trace Classifications*. A peak *Trace Classification* is a set of three scores that capture the shape and intensity of the traces associated with a single base call. (**a**) The classes and the criteria used to distinguish among them are listed and illustrated. A score from 0 to 1 is assigned for each of three classes that reflects the amount of *Strong* (*S*), *Medium* (*M*) and *Weak* (*W*) peak characteristic that is exhibited by the trace. (**b**) In this example, one of the four sets of traces is shown. The scores for the trace indicate a combination *Strong-Medium* peak.

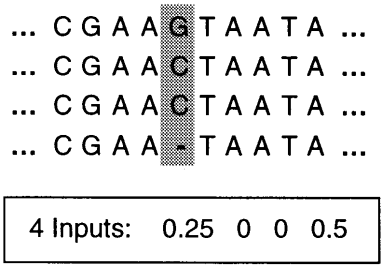


Fig. 4. *Base Call Fraction*. There are four aligned sequences in the highlighted column in this example. For each base call, we divide the number of their occurrences by the number of sequences. The *G* base call occurs once in four sequences, so its input is set to 0.25. Likewise, the inputs for *A*, *T* and *C* are 0, 0 and 0.5 (2 of 4), respectively.

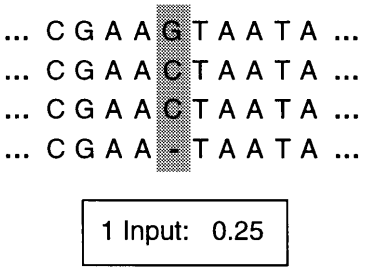


Fig. 5. *Gap Fraction*. For this example, we again have four aligned sequences in the highlighted column. For this input, we are only interested in gaps, so the single input is the number of gap occurrences divided by the number of sequences. Here a gap occurs once in four sequences, so the input is 0.25.

the traces. To do this, we employ *Trace Classification* scores described in Allex *et al*. (1996) and summarized in Figure 3.

The four input features we defined for an aligned column are listed next.

- *Base Call Fraction*
  The fraction of occurrences of G, A, T and C.
- *Gap Fraction*
  The fraction of occurrences of gaps.
- *Trace Peak Intensities*
  For each base, the trace peak intensity weighted by quality and averaged over the number of aligned sequences.
- *Trace Peak Shapes*
  For each base, the *Strong* (*S*) and *Medium* (*M*) *Trace Classification* scores weighted by quality and averaged over the number of aligned sequences.

Figures 4–7 contain the details of calculating the numerical inputs for these features.

We tested five network topologies. Each has five hidden units and five outputs. The desired outputs for the networks always consist of four zeros and a single one that represents either one of the four bases or a gap.

The input representations use combinations of the four possible input features described above. The simplest network, referred to as *Base Call*, uses an input representation that consists of the *Base Call Fraction* and the *Gap Fraction* features. The *Base Call* network is used as the control in testing our hypothesis that inputs that include trace information produce more accurate results than those that only consider base calls.

A second network, called *Trace Shape*, uses nine inputs that include the *Trace Peak Shapes* and *Gap Fraction* input features. A third network, *Trace Intensity*, has five inputs that use *Trace Peak Intensities* and *Gap Fraction* input features. The fourth network, referred to as *Trace Shape & Intensity*, uses both the *Trace Peak Intensities* and the *Trace Peak Shapes* as well as the *Gap Fraction* features in its 13 inputs.

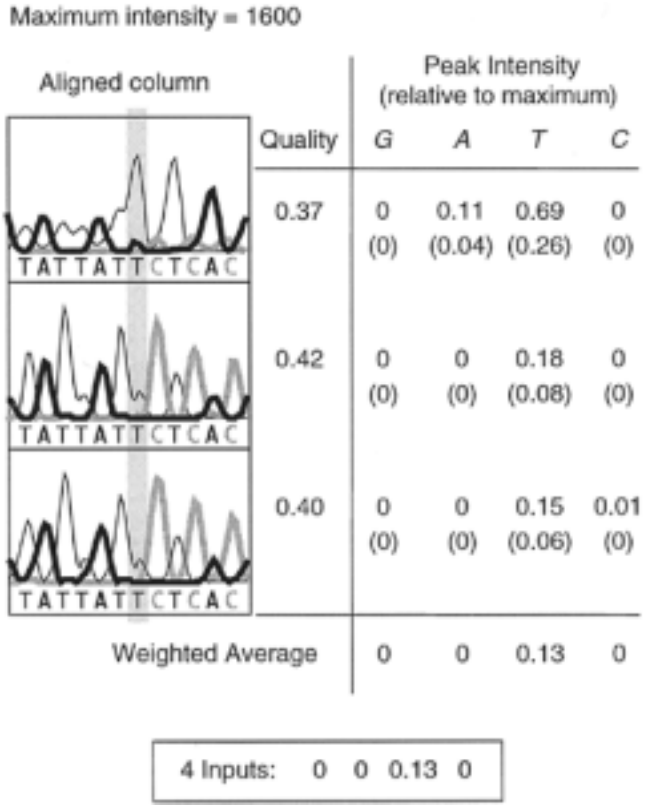


Fig. 6. *Trace Peak Intensities*. Three sequences are aligned in the highlighted column. For each of the four bases in each sequence, the intensity (value at the center of the column) of the trace is divided by the maximum possible trace value. This fraction is then multiplied by the quality value (Allex *et al*., 1997) assigned to the sequence. The average over the weighted values forms the input for each base. In this example, the maximum trace value is 1600 (a typical value for ABI traces). In the first sequence, the intensity of the *T* trace is 1104 and its intensity relative to the maximum is 0.69 (1104/1600). Values for all other bases in each sequence are calculated in the same way. The values are then weighted by the quality and the results are given in parentheses below each relative intensity. When averaged, the values yield the inputs 0, 0, 0.13 and 0.

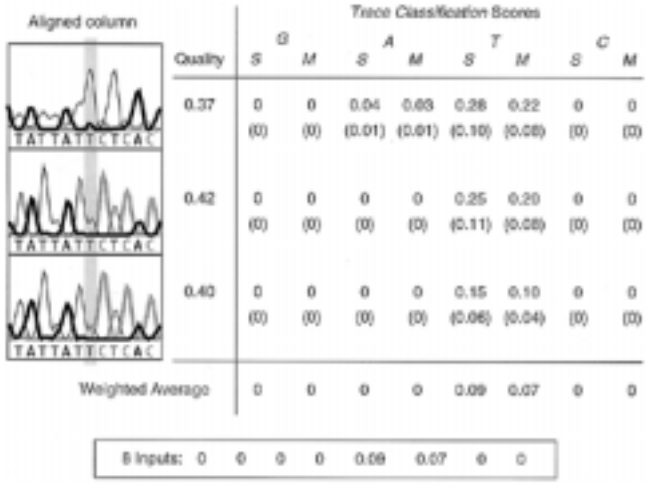


Fig. 7. *Trace Peak Shapes*. To form the inputs for the three aligned sequences in the highlighted column, we extract trace information using *Trace Classification* scores (Allex *et al*., 1996). We first compute the *Strong* (*S*) and *Medium* (*M*) peak scores for each of the four traces in each sequence. (We found *Weak* scores to be irrelevant and do not use them.) Each score is then multiplied by the quality score for its trace. The scores weighted by the quality are given in parentheses below the scores. There are two inputs for each base: the average over all the sequences of the weighted *Strong* scores and the average of the weighted *Medium* scores.

Finally, we tested one network that included all the possible input features: *Base Call*, *Trace Peak Intensities*, *Trace Peak Shapes* and *Gap Fraction*.

The five network topologies are summarized in Figure 8. To make a consensus call with one of these networks, we find the highest output value and its corresponding base or gap is the consensus call. Ambiguous calls may also be made by setting a threshold. If more than one output exceeds the threshold, then the appropriate ambiguous call is made. If only one output is above threshold, the call is unambiguous. In non-heterozygote DNA sequences, human editors resolve ambiguous calls to one of the four bases before submission to GenBank. Ambiguous calls serve to focus editors' attention on areas in the consensus that warrant closer examination. In the case of heterozygote genomes, ambiguous calls pinpoint differences between the alleles.

Implementation

We tested the effectiveness of the networks on examples with various distinct amounts of coverage (number of aligned sequences). Since almost any reasonable algorithm can make correct calls when the coverage is high, we believe that one criterion that can be used to identify a superior method is its accuracy even when the coverage is low. In addition, since every step required to sequence a fragment adds to the overall expense of sequencing, reducing the needed coverage means a substantial reduction in sequencing costs. In large sequencing projects, it is typical to produce a coverage of at least six in all areas to ensure accurate consensus sequences. This much coverage is not needed when using a method that is highly accurate with fewer aligned sequences.

To compare the input representations with varying amounts of coverage, we created example sets in which all of the examples for a particular set have the same coverage. We chose examples with coverages of two, three, four, five and six to form five sets. Each set contains 20 000 examples

Neural Network Name	# Inputs	Input Features
Base Call	5	• *Base Call Fraction* • *Gap Fraction*
Trace Shape	9	• *Trace Peak Shapes* • *Gap Fraction*
Trace Intensity	5	• *Trace Peak Intensities* • *Gap Fraction*
Trace Shape and Intensity	13	• *Trace Peak Shapes* • *Trace Peak Intensities* • *Gap Fraction*
All	17	• *Base Call Fraction* • *Trace Peak Shapes* • *Trace Peak Intensities* • *Gap Fraction*

Fig. 8. Network topologies. Each of the five networks has five hidden units and five outputs. The number of inputs ranges from five to 17.

of categorized data. Ten training and test sets are constructed from each example set such that each network is trained on 18 000 examples and tested on the remaining 2000. Each example occurs in exactly one test set and nine training sets disjoint from the test set. In these sets, examples with a desired output of *gap* are far outnumbered by examples with desired outputs of *G*, *A*, *T* or *C*. To enable the networks to learn to recognize gaps, gap examples are duplicated in the training sets so that they occur with about the same frequency as examples for each base. (Note that gap examples are not duplicated in test sets.)

The example sets are extracted from fragment assemblies of a 124 kb section of *Escherichia coli* supplied by the *E.coli* Genome Project at the University of Wisconsin (Blattner *et al*., 1997). The assemblies were created with DNASTAR's *SeqMan II* fragment assembly program. Although most of the data and alignments in the assemblies are quite good, sequence traces do vary in quality and some areas present more of a challenge for consensus calling. Figure 9 contains an example of an aligned region in one of the test assemblies that contains a fair amount of discrepancies, indicating imperfect underlying trace data and difficulties for consensus calling. The data and subsequent alignments included in our testing and results cover a wide range of quality from near perfect to quite inexact, as shown in Figure 9. Correct base calls used to categorize data are taken from *E.coli* sequences submitted to GenBank.

```
GCAANTAAAAANTGTTCCTTTGGGGTGAANANCCAAANATN-CCCNGCTGGGT
GCAATGAAATACTGTGCGT--GGGGTGAG-AGGCGAACATT-CCCGGCTGG--
GCAATGAAATATTATGCGN--GGGGTGAGAGGGCGAACATTCCCCGGCTGG--
GCAATGAAATACTGTNCGTN-GGGNTAAA-AGGC-AANNNTCCCCGGNNGG--
    ??   ? ? ? ? ????   ? ? ????? ?  ?????   ? ??  ??
```

Fig. 9. Test assembly alignment. The data used for testing are of varying quality. Displayed here is a region with four aligned sequences from one of the test assemblies. Columns whose base calls are not in total agreement are marked with a '?'. There is a fair amount of disagreement among the base calls, implying poorer quality underlying trace data. Consensus calling in this region is more difficult than in areas with near-perfect data.

NeuralWare Inc.'s *NeuralWorks Professional II* software was used for all neural network tests. We ran this software on an HP Pentium Pro 6/200 running Windows NT.

Discussion

We trained and tested each of the neural network topologies with the five example sets. For each coverage, we used 10-fold cross-validation and report accuracies averaged over the 10 test sets. During the training phase, each example in a training set was processed once.

Accuracy results for the five topologies are graphed in Figure 10. Of the five networks, we found that *Trace Shape & Intensity* produces the most accurate consensus calls. With a coverage of six, it makes only three errors in 20 000 calls. The range of accuracies is from 99.26% for a coverage of two to >99.98% with a coverage of six.

The network that uses only base call information in inputs, *Base Call*, has the lowest accuracies at every coverage. With two or three aligned sequences, this network has substantially poorer results than any of the other four networks. Except when the coverage is four sequences, differences between the *Base Call* and the *Trace Shape & Intensity* networks are statistically significant using a paired one-tailed *t*-test at the 95% confidence level. As with the other networks, the best results using the *Base Call* network are achieved when the coverage is six. With six aligned sequences, the error rate is eight in 20 000—more than double that of the best network that uses trace information.

In additional tests, we experimented with alternative plausible input representations. In one experiment, we extracted inputs from a broader context than a single column. Our premise was that the accuracy of the consensus calls could be increased by extending the inputs to include trace information for one or more bases 5′ to the base of interest. Parker *et al*. (1995) and Golden *et al*. (1993) have reported that intensity values for a base are affected by 5′ adjacent bases. For example, Parker et al. show that the intensity of a *C* peak following a *G* is relatively low. Several patterns such as these are described for dye–primer and dye–terminator labeled data (Parker *et al*., 1995; Perkin Elmer, 1995). We

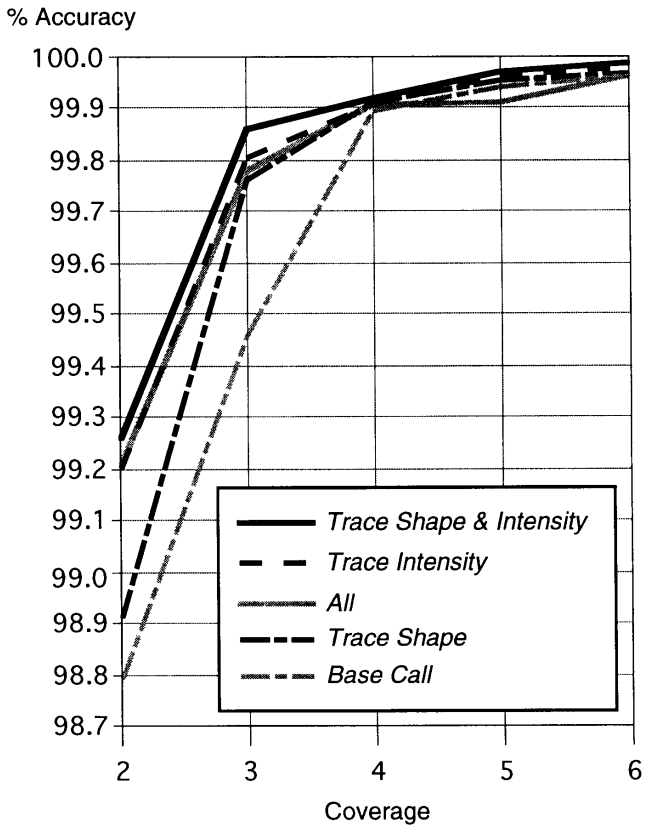


Fig. 10. Results. The *Trace Shape & Intensity* network produces the most accurate results at every coverage. With a coverage of four or more, the accuracies for all networks that use trace information are >99.9%.

believed that the neural networks could be trained to recognize these patterns, but in practice found no improvement in accuracy with the extended inputs.

In another experiment, we provided not just a single intensity input for each trace, but rather the intensities in a window surrounding the center of the base peaks. These are the same values that we use in calculating *Trace Classification* scores, but rather than transforming them algorithmically, we allow the network to process them. The network using this alternate input representation required more inputs, but yielded results very similar to the *Trace Shape & Intensity* network.

Our work demonstrates that neural networks can be an effective tool for determining the consensus of aligned DNA sequences. In particular, networks trained with input representations that use fluorescent trace information and ignore base calls are highly accurate. Further studies in utilizing traces in neural networks for consensus calling and related tasks are warranted.

References

Allex,C.F., Baldwin,S.F., Shavlik,J.W. and Blattner,F.R. (1996) Improving the quality of automatic DNA sequence assembly using fluorescent trace-data classifications. In *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology.* St Louis, MO. AAAI Press, Menlo Park, CA, pp. 3–14.

Allex,C.F., Baldwin,S.F., Shavlik,J.W. and Blattner,F.R. (1997) Increasing consensus accuracy in DNA fragment assemblies by incorporating fluorescent trace representations. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology.* Halkidiki, Greece. AAAI Press, pp. 3–14.

Ansorge,W., Sproat,B.S., Stegemann,J. and Schwager,C. (1986) A non-radioactive automated method for DNA sequence determination. *J. Biochem. Biophys. Methods*, **13**, 315–323.

Baldi,P and Brunak,S. (1998) *Bioinformatics: The Machine Learning Approach*. MIT Press, Cambridge, MA.

Blattner,F.R. *et al*. (1997) The complete genome sequence of *Escherichia coli* K-12. *Science*, **277**, 1453–1474.

Bonfield,J.K., Smith,K.F. and Staden,R. (1995) A new DNA sequence assembly program. *Nucleic Acids Res.*, **24**, 4992–4999.

Ewing,B. and Green,P. (1998) Base-calling of automated sequencer traces using Phred. II. Error probabilities. *Genome Res.*, **8**, 186–194.

Ewing,B., Hillier,L., Wendl,M.C. and Green,P. (1998) Base-calling of automated sequencer traces using Phred. I. Accuracy assessment. *Genome Res.*, **8**, 175–185.

Golden,J.B.,III, Torgersen,D. and Tibbetts,C. (1993) Pattern recognition for automated DNA sequencing: I. On-line signal conditioning and feature extraction for base calling. In *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology.* Bethesda, MD. AAAI Press, Menlo Park, CA, pp. 136–134.

Green,P. (1997) Genome sequence assembly. In *1st Annual Conference on Computational Genomics: Program and Abstract Book*. TIGR Science Education Foundation, Herndon, VA, p. 15.

Lawrence,C.B. and Solovyev,V.V. (1994) Assignment of position-specific error probability to primary DNA sequence data. *Nucleic Acids Res.*, **22**, 7.

McClelland,J.L. and Rumelhart,D.E. (1986) *Parallel Distributed Processing*. MIT Press, Cambridge, MA.

National Human Genome Research Institute (1998) NHGRI Standard for Quality of Human Genomic Sequence. http://www.nhgri.nih.gov/Grant_info/Funding/Statements/RFA/quality_standard.html.

Parker,L.T., Deng,Q., Zakeri,H., Carlson,C., Nickerson,D.A. and Kwok,P.Y. (1995) Peak height variations in automated sequencing of PCR products using taq dye-terminator chemistry. *BioTechniques*, **19**, 116–121.

Perkin Elmer (1995) *DNA Sequencing: Chemistry Guide*. Foster City, CA.

Smith,L.M., Sanders,J.Z., Kaiser,R.J., Hughes,P., Dodd,C., Connell,C.R., Heiner,C., Kent,S.B.H. and Hood,L.E. (1986) Fluorescence detection in automated DNA sequence analysis. *Nature*, **321**, 674–679.

Sutton,G., White,O., Adams,M. and Kerlavage,A. (1995) TIGR Assembler: A new tool for assembling large shotgun sequencing projects. *Genome Sci. Technol.*, **1**, 9–19.