Machine Learning in Basecalling – Decoding Trace Peak Behaviour

David Thornley and Stavros Petridis, *Members, IEEE*

***Abstract*— DNA sequence basecalling is commonly regarded as a solved problem, despite significant error rates being reflected in inaccuracies in databases and genome annotations. These errors commonly arise from an inability to sequence through peak height variations in DNA sequencing traces from the Sanger sequencing method. Recent efforts toward improving basecalling accuracy have taken the form of more sophisticated digital filters and feature detectors. We demonstrate that the variation in peak heights itself encodes novel information which can be used for basecalling. To isolate this information for a clear demonstration, we perform a peculiar blind basecalling experiment using ABI processed output. Using classifiers responding to measurements in the context of the basecalling position, we call bases without reference to the peak heights at the basecalling position itself. Tree classifiers indicate which features are pertinent, and the application of neural nets to these features results in a startlingly high initial success rate of 78%. Our analysis indicates that we can make viable basecalls using information that has never been accessed before.**

I. Introduction

After almost thirty years of development, DNA sequencing using the Sanger method remains the dominant approach [1], [2], [3], but it is still subject to errors [4] and restricted read lengths. Up to 1% errors in the "high confidence" region of a trace are not uncommon. Miscalls or indels are commonly recovered by multiple coverage of the region of DNA being sequenced. Alternative approaches are required when addressing an individual diagnostic target. A new means for reducing error rates or increasing read lengths will reduce the depth of coverage required in sequencing applications, and hence reduce resource consumption.

Early basecallers, such as that produced by ABI and distributed with their equipment, performed pre-processing steps to simplify the data. This 'analysed' trace data enabled a simple basecalling method based on finding the largest peak at each position. Later methods, such as PHRED [3], [4] calibrated a model of peak spacing to assess trace peaks against expected position. This improved error rates on many systems with even peak spacing, but "without such peak spacing normalization, PHRED may predict peaks poorly in some regions of the trace, resulting in increased numbers of indels" [3]. PHRED provides confidence measures for each basecall in the form of a decibel quality number expressing the degree to which the peak for that basecall corresponds to the position predicted by the calibrated curve. We know [5] that there is detail in the position of peaks, varying from a calibrated curve, which correlates with sequence composition. This may impact the accuracy of PHRED's confidence measure, since the calibrated curve cannot match this behaviour.

In 1994, Lipshutz et al [6] demonstrated that a legitimate confidence measure could be estimated using the CART tree classifier [7], basing its decisions on the bases and trace data around the basecall to which confidence is to be ascribed. We reproduced Lipshutz' analysis of variance (ANOVA) analyses on our dye terminator data, and the results were practically indistinguishable from theirs, despite their pertaining to dye primer data. In 1997, Thornley [8] proposed a mechanism for the variation we see in peak heights due to base sequence specific enzyme selectivity. This was accompanied by a method for exploiting peak height variation given a suitable model. This method was applied to a small data set for a basic demonstration of the presence of basecalling evidence in contextual peak heights. The demonstration took the form of analysing the contextual peak heights *excluding the peaks at the calling position*, to give a "blind-spot" basecalling success rate of 42%. This was considered encouraging, as it compared favourably with the 25% expected from random guessing.

Our research into the use of the context information surrounding the basecalling position, or "pivot" as we refer to it for convenience, comprises modeling the Sanger reaction and sequencing equipment in detail. We aim to use machine learning techniques to guide our inquiries into contextual behaviour. Two of these techniques, the tree classifier and the artificial neural net, have allowed us to provide an indication of the potential efficacy of this information, and to begin to learn its form.

We provide some blind basecalling results to illustrate the potential in the contextual information. In fact, we demonstrate that viable basecalls are encoded in the context. Integrating this information into a full basecaller which can address the lower quality data later in the trace is a research question that we are currently addressing [9].

In the work described here, we achieve a success rate in blind-spot basecalling of 78% using machine-learning tools [10], [11] operating on recent genomic sequencing data. The measurements we use are ostensibly the same as those used by Lipshutz *et al*. However, after assembling tuples of basecalls and peak heights (and simple functions of them),

David Thornley is employed on EPSRC Grant GR/S60266/01.

D. Thornley is with the Department of Computing, Imperial College London, UK. (e-mail: djt@doc.ic.ac.uk).

S..Petridis was an MSc project student in the Summer of 2005 supervised by Thornley at Imperial College London. He is now with the Robotics Institute at Carnegie Mellon University, USA (e-mail: sp104@alumni.doc.ic.ac.uk).

we train classifiers (tree classifiers and artificial neural nets) on them to attempt to reproduce the correct basecalls as the output class. An interesting addition to the classification variable set is peak spacing, as it illustrates the importance of understanding the difference between correlation and causality, as discussed in the analysis section. Our most important subtraction is the peak data at the basecalling position. Thus, this is a unique result demonstrating the direct use of context information which has not previously been used to call a base.

II. METHODS

A. Data Processing

The data we use come from a genomic sequencing trace data set used in quality control testing at the Wellcome Trust Sanger Institute.

We perform a skyline normalization [12] of the data to remove the overall decay in peak heights, and the initial rise thought to be due to length dependent loading efficiency, by fitting a quadratic curve to the apexes of the peaks and dividing through by the value of the quadratic at each peak sample point. Figure 1 shows an example of this for the T lane.

Identifying high quality regions of trace data is a largely open question. We employed a highly conservative approach of searching from the centre of the trace outwards for the first failed basecall to the left and right, then backing off toward the centre by 80 base positions. This gives us a region of relatively high quality basecalls and trace data. This decision is lent credence by the success rate of the trivial basecalling classifier shown in Figure 2. The goal of the present work is to demonstrate the presence of novel context information, and not yet to process through poor data, so we do not include it.

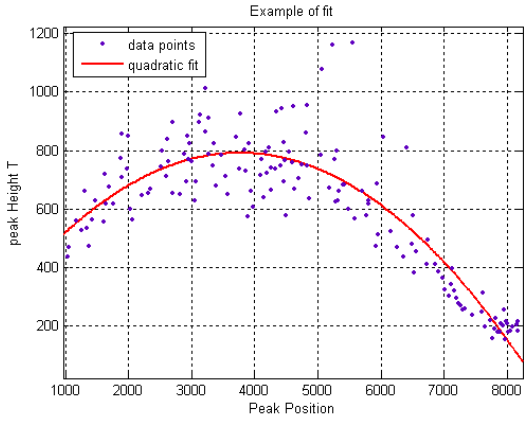

Fig 1: Example of a skyline fit to a lane of trace data peak apeces. We perform this independently for each logical lane since there is no guarantee that any relationship between the lanes is retained in ABI's processing steps.

The normalized data was split into overlapping 5mer (5 contiguous bases) windows of basecalls and trace data. We refer to the 5 base sequence in this window as the "footprint" of the sample. We record the height of the trace in each of the four lanes at each base position corresponding to the basecalls, along with the sample position of those basecalls, and additional calculated features such as spacing and ratios. Table 1 below shows the ensemble of features we include in the tuple when the basecall at position 2 within the window is to be given by the output class. The signal heights at the pivot are excluded in all but two experiments, which were performed to assess the quality of the selected data.

Features	Description
Bases	These are the basecalls in the window except at the pivot, so these correspond to abi1, abi3, abi4, abi5 in Lipshutz *et al*.
Peak Heights	x1, x3,x4,x5, where x is a, c, g or t
Peak Heights Ratios	If 5mer is ATGCA then ratios are defined as r1=a1/(a1+t3), r3=t3/(t3+g4), r4=g4/(g4+a5)
Peak Spacing	p3-p1, p4-p3, p5-p4

Table 1: Features made available to the classifiers. The labelling is identical to that used in Lipshutz *et al*, with the addition of *p* for each base in the window for calculating peak spacings.

For this proof of concept work, we have therefore performed minimal processing in order to exercise the classification approach in as un-biased as possible a manner.

B. Tree Classifiers

A full tree is grown using the training data, which is then pruned to minimize misclassification in the validation data. We assessed a range of splitting criteria for the classification trees, and the Gini impurity [7] measure performed best, with a node impurity of 40 providing the best compromise between performance and computational time. In order to avoid overfitting we pruned all the trees using the reduced-error pruning method against the validation set.

We describe two types of tree classification architecture as detailed below [13].

Type 1: In this type, we produce a single *four-class-one-classifier* tree which selects the class A, C, G or T based on input data. An example of type 1 is shown in Figure 2 (the only one we generated which is small enough to be susceptible to analysis in the static format of a paper). This tree is the result of giving the classifier the peak heights at the pivot, as well as all the contextual variables we use in the main body of experimentation. Note that the result does not query any context information. The error rate of this tree is 0.024%.

This gives a handle on a baseline accuracy we might hope for without correcting the sequence against a consensus. In this proof-of-concept work, we have been conservative with the data, rather than take every base indicated as correct by a consensus. In our main results section, we also show the

result of giving the classifier ratios of the peak heights at the pivot. This result is even better, and we believe this is because the classifier is in a position to perform exactly the peak height comparisons we would use in a classic basecaller, rather than having to construct an approximation to the partitioning of the space of these comparisons.

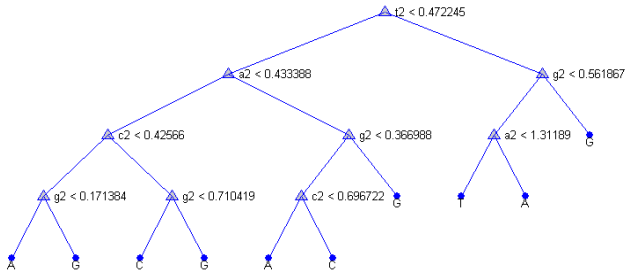

Fig. 2: Tree constructed when pivot heights are also included in the input patterns. The branch to the left is taken when the predicate is satisfied.

Type 2: In our second type of classification regime, four *one-class-one-classifier* modules are combined in a parallel architecture and their output is combined using a voter. Ideally, only one classifier will vote yes and the other three will vote no. However, it is common that more than one module will vote yes or all of them will vote no. There are several techniques to solve this ambiguity, depending on the way each module is implemented. We use the statistical probability associated with the True or False classifications. If there is more than one True classification, the classification with the highest probability for True is output. If there are no True classifications, then the class with the lowest False probability is output.

Table 2 shows the error rate of the four type-2 classifiers.

	A	C	G	T
Error rate	17.77%	14.94%	11.03%	19.03%

Table 2: Error rate of recognising a single class in a component for a type 2 or type 3 tree classifier

Note that these error rates look small when compared to the figure of approximately 22% we achieve in the results section. This reflects the fact that these classifiers individually only have to distinguish between the two classes True or False.

C. *Artificial Neural Nets*

Artificial neural nets [14] are powerful classifiers, but are commonly highly domain-specific. Our results are therefore only guaranteed meaningful when the training and test data are independent.

Only feed-forward neural networks with one hidden layer and a four-output layer were used. We chose a hyperbolic tangent activation function for the hidden and output layers. We selected values of +1 for True, and –1 as False at the outputs, and hence the targets for training: for example the output $[-1\ -1\ +1\ -1]^T$ means that the output pattern is to be assigned to class G (the first output represents A, the second C &c.). This architecture outputs for each of the four bases, so we refer to this as a type 1 classifier in a similar manner to the tree classifiers.

The resilient backpropagation training algorithm is used in this exploratory work for its speed of operation [15]. Most of the peak heights take values between 0 and 5, which already lie in a useful region of the domain of the hyperbolic tangent, so the inputs were not further manipulated. Using the mean square for the error function, the choice of 70 hidden units was found by a trial-and-error to provide a good balance between efficacy and computation cost. Each network was trained for 4000 epochs and the learning rate was set to 0.005.

D. *Experimental Procedure*

A suite of training and test data with ten independent sets of each was created to enable the different classification regimes to be exercised in a controlled manner. Each training set contains a number of 5mers varying from 28754 to 137845 samples with a total of 593836 samples. The number of 5mers in the 10 test sets varies from 29757 to 113090 making a total of 603011. To maximize the number of independent training and test pairings possible, a single validation set of 55782 5mers was used to prune all trees.

As well as using single classifiers, we have briefly explored the use of the combination of a number classifiers trained on different data [16]. We performed “bagging” of multiple classifiers trained on disjoint sets of data by either voting or signal accumulation.

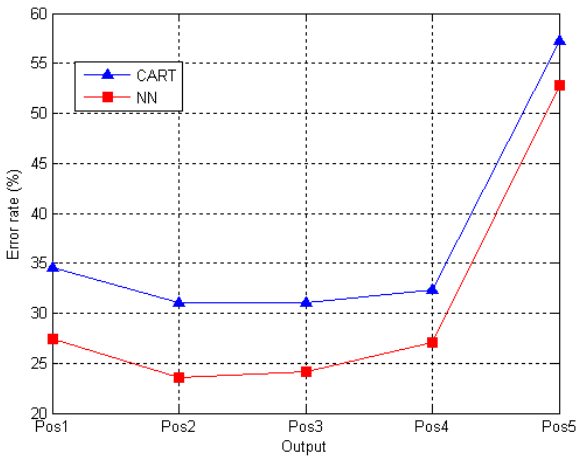

Fig. 3: Error rate for each classification method and for each output position in type 1 classifiers.

In the experiments we present in this paper, we use the basecall at position 2 in the window as the output class because this gives the highest accuracy of blind basecalling as shown in Figure 3.

III. Results

A. Data Processing

The distribution of sequence footprints (5-mers) we gather affects the meaning of our core result. Figure 4 shows their approximate distribution. Each bar shows an ensemble of sixteen 5mers to simplify the graph. This shows that in this data, in common with any genomic DNA set, the distribution of sequence features is not uniformly random. Every data set contains at least two instances of each 5mer. We include experimental results which illustrate the advantageous effect of this distribution on basecalling rates.

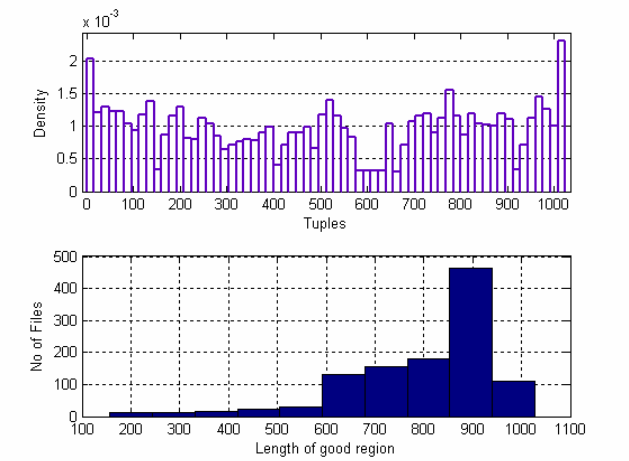

Fig. 4: Top: probability histogram of seeing each 5mer. Bottom: we also show the distribution of trace lengths in test set 1 to demonstrate that the read lengths we accept are not unusually short or long.

B. Feature Selection

In order to evaluate the features and find the most appropriate for basecalling we use tree classifiers. Table 3 shows the results of applying tree classifiers of type 1 and 2 to data set 1. For type 1, we used a training set of 62173 samples, a validation set of 55782 samples, and a test set of 74629 samples. In type 2, we used the same training and validation data for each of the four trees in a single classifier, but different sets across the classifiers in a bagged group.

Table 4 shows results of bagging ten and nineteen type 1 tree classifiers, the latter exercising the full set of training data, tested against one test set (to leave the remainder as training data for the larger bagging example). Table 5 shows results from application of type 1 neural networks, and of bagging ten such neural networks.

These experiments were carried out on a commodity desktop PC, and some of the bagging training runs lasted on the order of a day, so some elements of table 3, 4 and 5 are not populated. Trends in the results show that the missing figures would not represent key results, except those in table 4, where there is a clear advantage to bagging more trees, and adding more features, which result in longer training times.

Tree classifiers	Type 1	Type 2
Bases	65.73%	68.78%
Heights	33.33%	32.82%
Heights + Bases	33.37%	32.52%
Heights + Pivot Heights + Bases	0.024%	0.029%
Heights + Pivot Ratios	0.0067%	-
Spacing	60.61%	61.34%
Heights + Spacing	31.01%	30.43%
Ratios	55.37%	55.98%
Heights + Ratios	33.79%	32.97%
Spacing + Ratios	48.95%	50.05%
Heights + Ratios + Bases	33.79%	32.87%
Heights + Ratios + Spacing	30.69%	29.76%
Heights + Spacing + Bases	31.09%	30.18%

Table 3: Error rates of blind spot basecalling using tree classifiers on a range of variable sets.

Bagged type 1 trees	Bagging(10 trees)	Bagging(19 trees)
Heights Only	29.59%	27.93%
Heights+Spacing	27.17%	-
Heights+Ratios+Spacing	26.63%	-

Table 4: Multiple bagged trees.

Neural nets	Type 1	Bagging (10 nets)
Heights Only	26.91%	25.74%
Heights+Spacing	23.52%	22.10%
Heights+Ratios+Spacing	23.42%	-

Table 5: The results of applying a single neural net, and 10 bagged neural nets to test set 1.

	Combination Rule:		Majority	Sum
	tree-Type1	NN-Type1	NN-Ensemble	NN-Ensemble
Test1	31.011	23.522	21.281	21.098
Test2	30.686	23.918	21.424	21.14
Test3	30.139	22.274	20.941	20.802
Test4	31.414	23.789	21.427	21.198
Test5	31.998	24.522	22.286	22.158
Test6	30.299	22.767	20.123	19.889
Test7	31.63	24.131	22.075	21.837
Test8	34.993	27.906	26.353	26.179
Test9	33.067	25.535	23.626	23.496
Test10	31.919	24.664	22.433	22.185
Mean	31.7156	24.3028	22.1969	**21.9982**
Variance	2.09289	2.46731	3.03349	3.0973

Table 6: Variation in tree and neural net (NN) classifier error rates

C. *Variation over Training/Test Pairs*

Table 6 show the variation in error rate for type 1 classifiers implemented using trees and neural nets, and for bagged neural nets using either a single vote for the preferred base per net (these being summed to select the majority class for output), or using the sum of outputs for each base from all nets to select that with the largest total. The variable set in each case contains peak heights and spacing, and this is true of the examples in the following sections for consistency.

D. *First and Second Choice Performance*

If we define a 2nd choice failure rate as the correct base not falling in the classifiers' first or second choices, we see the success rates in table 7.

	2nd Choice	Sum-2nd Choice
	type1 NN	NN-Ensemble
Test1	7.0723	5.9615
Test2	7.1037	5.8951
Test3	6.5147	5.7404
Test4	6.9415	5.7837
Test5	7.7217	6.3193
Test6	6.433	5.3309
Test7	7.3234	6.2331
Test8	9.3927	8.2031
Test9	7.9092	6.8277
Test10	7.5046	6.4271
Mean	7.39168	**6.27219**
Variance	0.719333	0.634547

Table 7: Variation in rates of "missing" the correct basecall with first and second choice candidates for neural net (NN) classifiers.

E. *Confusion Matrices*

Confusion matrices give an indication of the stability of the classification. An example *1-tree-4-class* tree classifier's confusion matrix is given in table 8. The base assigned by the classifier is given a column, and the row corresponds to the 'true' base at that position. Each table entry therefore represents the percentage of the row's base being assigned the column's class by the classifier.

Tree	Assigned			
True	A	C	G	T
A	72.992	8.97	5.5647	12.474
C	17.8	58.83	6.4145	16.955
G	6.7303	5.8384	73.553	13.878
T	12.005	10.653	8.2517	69.091

Table 8: A type 1 classification tree confusion matrix. The column gives the correct base, and the row the base assigned by the classifier.

An example of a similar confusion matrix for a four-output neural net is given in table 9, followed by a bar chart of the correct percentages for each base for the two forms of classifier (red for tree, blue for neural net) in Figure 5.

Net	Assigned			
True	A	C	G	T
A	80.917	7.8973	3.4053	7.7802
C	14.406	65.62	5.0858	14.889
G	5.0098	4.4532	79.069	11.468
T	7.9849	8.4679	5.4551	78.092

Table 9: A neural net confusion matrix arranged as table 8.

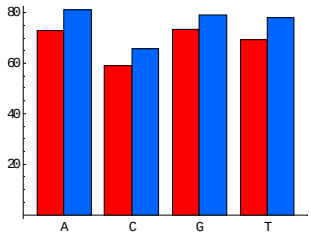

Figure 5: Percentages of correct ascription of class when the correct basecall is each of A, C, G and T. Red bars show results for a tree classifier, and blue for a neural net.

F. *Confidence*

Lipshutz *et al* [6] estimated the confidence in basecalls by removing the detail of peak height variation in their data by normalizing the peaks at each position such that the largest had unit height. This therefore essentially measured the local "noise" levels. We suggest that the peak height behaviour must be explicitly captured to properly assess confidence, and this will be addressed in further work.

The leaves of the basecall classification tree contain the probabilities of each of the potential output classes. A wide range of functions of this ensemble of such statistical probabilities could be formulated, but we show the raw highest probability, and perhaps the most intuitive second option in which the difference between the highest and next highest probabilities is used.

The bar charts in Figures 7 and 8 show the total number of correct and incorrect basecalls using a type 1 tree classifier in bins of an interval of 0.025 in confidence. Blue bars show the number of correct basecalls with the indicated confidence estimate, red bars show incorrect basecalls, and the purple regions are simply the bars overlapping.

IV. ANALYSIS

A. *Classifiers*

Response to peak height information is clearly advantageous. Initially we use tree classifiers, which are quicker to train, and after establishing the best combination of variables, we then use neural nets, which are commonly powerful discriminators.

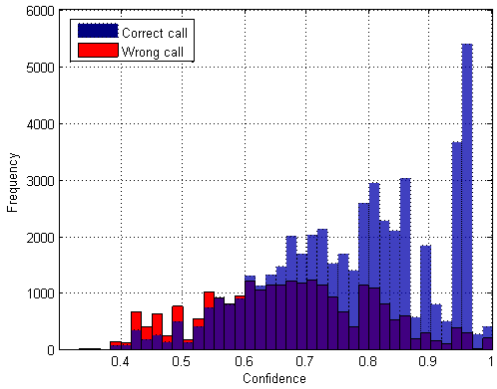

Figure 6: Confidence values, when confidence is defined as the maximum probability. Blue bars show the frequency of correct calls ascribed a given confidence, and red incorrect calls.

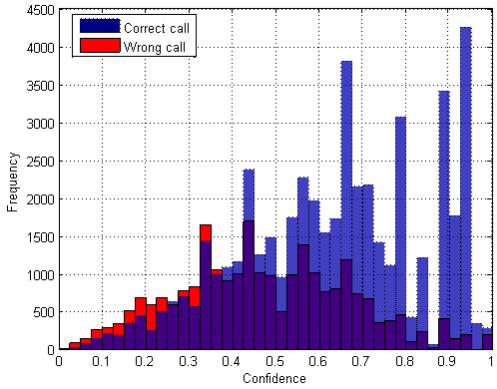

Figure 7: Confidence estimates when confidence is defined as the difference between the two highest probabilities (first-minus-second).

Sequence bias: Before looking into the results of accessing peak height information, we must point out that if we provide the classifier with only the surrounding basecalls, our error rate is as low 65.73% (table 3). This must be compared with the 75% error rate we would expect from making a random guess. This difference arises from the fact that the footprints in the data we use are not evenly distributed. Thus, we have an *a priori* advantage in basecalling using context information before we start using peak heights. This advantage will be different for each organism and each sequencing effort, simply because the distribution of sequence motifs will vary. Thus, our baseline success rate should be taken to lie between 25% and 34.27%. This makes the final result look less impressive than it might otherwise.

Spacing: If we give the classifier just the peak spacing, we have an error rate of only 60.61% (see table 3), again compared with 75%. This is a success rate of almost 40%. There are two main possibilities for the source of the advantage in having a peak spacing measure. It may be that the preprocessing stage of the ABI data gathering process places the peaks from A, C, G and T on slightly differently aligned progressions. Since we are using peak positions which correspond to the base called by ABI's software, this would provide explicit information as to the sequence composition.

Another source of information may be in the sequence dependent mobility of the DNA fragments. A correlation between the two 3' bases of a DNA fragment and its mobility was demonstrated by Bowling *et al* [5].

When peak height information is given to the classifier as well as peak spacing and sample number positions, decisions based on peak spacing appear near the root of the tree in a similar manner to those dependent on peak position in Lipshutz work. We conclude that the spacing, which increases gradually during the progress of the trace, is being used as an indicator of progress in the trace in preference to sample number. The trace data may begin at an arbitrary sample number, but the relationship between peak width spread and increase in peak spacing over the trace is probably well preserved, we favour this explanation.

There are some decisions made based on spacing lower down the tree, and this may indeed be responding to detail. The change in success rate when peak spacing is included in the variables in addition to peak height information is smaller than the isolated advantage of the peak spacing – with peak heights it is (33.33-31.01)=2.32% compared to a figure without peak heights of (75-60.61)=14.39% (see table 3). The use of peak spacing information merits an independent investigation which is beyond the scope of this present report. For example, Lipshutz identifies progress through the trace with reducing confidence, whereas we believe our results suggest that the reaction characteristics change over the progress of the reaction along the template DNA in a manner which can be tracked.

Pivot peak data: In one experiment (table 3), we gave the tree classification routines access to the peak heights at the pivot. This resulted in the tree shown in figure 2, which does not query any other variables. This is because the pivot data provides sufficient information to call the bases in this high quality data.

Best tree classifiers: The basic type 1 classification regime implemented using a tree achieves a maximum success rate of 100-30.69=**69.31%** (table 3) using peak heights, peak ratios, and peak spacing measurements. We believe that the inclusion of peak height ratios simply makes available some more concise queries, rather than adding information. Bagging several trees is highly advantageous, as shown in table 4, where the success rate was as high as **73.37%**. Bagging 19 trees was better than bagging 10 trees for peak height data alone, so we speculate that working on a system with the resources to train 19 trees for the best variable set would further improve the results.

The apparent advantage in including peak height ratios (table 3) in the variable set is smaller than the variation between data sets (see table 6). The only ratios which make a difference are in the isolated experiment to construct a trivial classifier, where the use of ratios between peak heights in the four lanes at the pivot allowed the classifier to make direct comparisons, rather than making approximations, as in figure 2.

Our preferred explanation for the slight advantage in including peak height ratios is that the skyline normalization is flawed in a systematic manner, corresponding to features which erroneously dip below the skyline or rise above the skyline, hence having "erroneous" absolute values. The fact that most of the decisions are based on absolute peak sizes suggests these situations are rare.

Neural nets: Shifting our attention from classification trees to more powerful neural networks, we find an across-the-board increase in success rates. Table 5 shows the higher overall success rate than trees, and a similar variation across the variable ensembles used in the classifier.

The best result comes from using peak heights and spacing, achieving a blind spot basecalling success rate of **78.01%**. (see table 6) (Note that neural network training involves a random element, and we show the mean of the ten experiments.) This compares well with out highest lower bound on success rate of 34.27% when we do not have access to the peak data. The peak spacing information seems to be more advantageous to the neural network approach. Perhaps the detailed response in mobility to sequence composition is being used – we can not tell without extensive experimentation with the sensitivity of the outputs with respect to perturbation of the inputs.

The confusion matrices in tables 8 and 9, and the bar chart in Figure 5 give some indication of the stability of the results. C and T bases are clearly the least well predicted. However, the confusion matrix result speaks to a more complex behaviour in which the implications of a C or T's presence in the footprint are less well characterized. This means that a C or T at the pivot has an effect on the surrounding peak heights (to which the classifier responds) which is less stable than the effect of an A or a G.

It is notable that a DNA sequencing trace from the complementary strand would have an G or A in place of the C or T respectively, so that the clearer information is a available in remarkably symmetric manner. This means that if the trace data for a given base is obscured on the forward and reverse strands from a given double stranded DNA source, one of the strands will have strong information.

Second choice: The results in table 7 show us that the application of classifiers can home in on the most likely basecall further. A success rate of **93.73%** of the correct besecall falling in the first or second choices of a bagged neural net classifier is tantalizing. This is hard to quantify in terms of individual basecalls, but in genomic sequencing where multiple traces contribute to each base position, this probabilistic information is desirable.

B. Confidence

We use the class with the highest probability as the output, so the simplest confidence estimate is provided by its statistical probability as indicated in the leaf of a tree classifier. Figure 6 shows this confidence estimate for the wrong (red) and correct calls (blue) from a type 1 tree classifier. The overlapping area is shown in purple. Confidence estimates less than 0.4 are absent, relating perhaps to the node purity cutoff, so that region is not shown.

A more discriminatory estimate is provided by taking the difference between the highest probability and the next higher probability as shown in Figure 7. In both Figures 7 and 8 we see that confidence estimate values close 1 give us a truly high probability of a correct call. When the second measure is 0.9 or higher, for example, we can infer high confidence in the result. We can define a function from the tree node class probabilities to give a measure calculated on the discrimination indicated by this diagram. The PHRED quality measure for basecalls which are in error with frequency 0.22 (corresponding to our success rate of 78%) is 6.27. For our blinded basecalls with a first-minus-second confidence estimate greater than 0.9, this quality measure is over 10, and hence above the common low quality threshhold, without using the information relied upon by all other basecallers.

V. CONCLUSIONS

The use of computational intelligence has enabled a unique demonstration of the use of context information in DNA sequencing traces described in [8] for calling bases. This work arose as a thread in our research pursuing a model of the sequencing process to be used as part of an abduction approach to basecalling [9]. Classification tools therefore serve the dual purpose of providing basecalling functionality, and informing the synthesis of phenomenological models.

The tree classifiers we generated suggest that the reaction behaviour changes with progress through the reaction. To our knowledge, this is the only manner by which this might have been efficiently discovered. Note also that this change occurs well before the apparent degradation of the data later in the trace. We will be using this approach in our research to investigate further candidate features which may increase the discrimination power of the accessible context information.

We have demonstrated that the variation in peak heights in DNA sequencing traces from the Sanger sequencing method encodes information which can be used for basecalling. This was achieved by using a peculiar blind basecalling experiment to demonstrate the potential for advantage to be gained from using a classifier to respond to this variation. This has produced some acceptable basecalls, despite the significant approximations made in normalizing and linearizing the problem. When integrated with pivot data using our hypothesis rejection approach [8], we expect this information to reduce the error rates of modern basecallers.

Fusion of contextual information with the prima fasciae evidence of a dominant signal peak at the pivot may be achieved by a variety of means. An example experiment will be to use a classifier based on that of Lipshutz, which classifies existing basecalls as true or false, and this can be used as an indicator of whether to use our blinded classifier. It will be interesting to see if Lipshutz' choice of the fifth position in a trace window of five positions is optimal for

assessment of confidence in current data, especially if we do not normalize-out the peak height pattern.

This work with classifiers is part of our ongoing investigation of the use of contextual information for basecalling. Further work on the use of such tools for this purpose includes the application of more sophisticated classifiers, such as fuzzy trees, deeper neural networks, and support vector machines. Machine learning will also guide our enhancement of phenomenological models of the processes involved in DNA sequencing, which will initially complement, and eventually replace classification and regression elements. The selection of features we attempt to model is guided to an extent by their apparent significance in classification and regression experiments, and this information is guiding our experiment design.

REFERENCES

[1] F. Sanger, S. Nicklen and A.R. Coulson. DNA sequencing with chain terminator inhibitors. *Proc. Natl. Acad. Sci.* 74, 5463 5467, 1977d ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] C. Connel, S. Fung, C. Heiner, J. Bridgham, V. Chakerian, E. Heron, B. Jones, S. Menchen, W. Mordan, M. Raff, M. Recknor, L. Smith, J. Springer, S. Woo and M. Hunkapiller. (1987) Automated DNA Sequence Analysis *BioTechniques* **5**, 342-348. H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.

[3] Ewing B, Hillier L, Wendl MC, Green P, "Basecalling of automated sequencer traces using phred. I. Accuracy assessment", GENOME RESEARCH 8 (3): 175-185 MAR 1998 E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *IEEE Trans. Antennas Propagat.*, to be published.

[4] Ewing B, Green P, "Basecalling of automated sequencer traces using phred. II. Error probabilities", GENOME RESEARCH 8 (3): 186-194 MAR 1998 C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[5] John M. Bowling, Kaylon L. Bruner, Joan L. Cmarik and Clark Tibbetts.(1991) "Neighbouring nucleotide interactions during DNA sequencing gel electrophoresis", *Nucleic Acids Research* **19**, 3089-3097.

[6] Robert J. Lipschutz, Fred Taverner, Kevin Hennesy, George Hartzell and Ron Davis. DNA sequence confidence estimation. *Genomics* 19, 417 – 424 (1994)

[7] L. Breiman, J. H. Friedman, R.A. Olshen and C. J. Stone. *Classification and regression trees*, Wadsworth, Inc., Belmont, California, 1984.

[8] D. J. Thornley. "Analysis of trace data from fluorescence based Sanger sequencing". PhD thesis, University of London, Imperial College of Science, Technology and Medicine, Department of Computing, 1997.

[9] International Patent Application WO96/20286 July 4, 1996, European Patent EP0799320 Mar. 7 2001 and US Patent 6,090,550, Jul. 18, 2000

[10] Tom M. Mitchell. *Machine Learning.* McGraw-Hill 1997

[11] Richard O. Duda, Peter E. Stork, David G. Stork. *Pattern Classification.* WILEY – INTERSCIENCE, N.Y. 2001

[12] Lucio Andrade, Elias S. Manolakos. Skyline Normalization of DNA Chromatograms by Regression, *in Workshop On Genomic Signal Processing and Statistics (GENSIPS)*, 2002, pp.CP2—7:1-4

[13] Manish Sarkar. Modular Pattern Classifiers: A Brief Survey in Proc. IEEE Int. Conf. Systems, Man and Cybernetics, vol. 4, 2000, pp.2878-2883.

[14] A.K. Jain, Jianchang Mao, K.M. mohiuddin, Artificial Neural Networks: a tutorial, *IEEE Computer Mag.*, pp 31-44, Mar 1996

[15] Riedmiller, M., and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm", Proceedings of the IEEE International Conference on Neural Networks, 1993

[16] David Opitz, Richard Maclin. Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research 11* (1999) 169-198