5-18-2007

Analysis of Nanopore Detector Measurements using Machine Learning Methods, with Application to Single-Molecule Kinetics

Matthew Landry
University of New Orleans

Analysis of Nanopore Detector Measurements using Machine Learning Methods, with Application to Single-Molecule Kinetics

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Master of Science
in
Computer Science

by

Matthew Landry

B.S. University of New Orleans, 2005
M.S. University of New Orleans, 2007

May 2007

Table of Contents

List of Figures iii
Abstract iv
Introduction 1
Background 2
    Nanopore Detector 2
    Cheminformatics 2
    HMM Feature Extraction 4
    AdaBoost 6
    SVM Classification 6
Results 8
    Emission Inversion 8
    Spike Analysis 11
    Dwell Time Analysis 12
    Feature Selection 14
Discussion 16
    Emission Inversion 16
    Spike Analysis 16
    Dwell Time Analysis 16
    Feature Selection 17
Conclusions 19
Methods 20
    Emission Inversion 20
    Dwell Time Analysis 20
    Feature Selection 21
Acknowledgements 23
References 24
Vita 26

List of Figures

Figure 1 2
Figure 2 3
Figure 3 4
Figure 4 6
Figure 5 7
Figure 6 8
Figure 7a 10
Figure 7b 10
Figure 7c 10
Figure 8 11
Figure 9 11
Figure 10 12
Figure 11 12
Figure 12 13
Figure 13 13
Figure 14 14
Figure 15 15
Figure 16 15
Figure 17 15
Figure 18 18

Abstract

At its core, a nanopore detector has a nanometer-scale biological membrane across which a voltage is applied. The voltage draws a DNA molecule into an α-hemolysin channel in the membrane. Consequently, a distinctive channel current blockade signal is created as the molecule flexes and interacts with the channel. This flexing of the molecule is characterized by different blockade levels in the channel current signal. Previous experiments have shown that a nanopore detector is sufficiently sensitive such that nearly identical DNA molecules were classified successfully using machine learning techniques such as Hidden Markov Models and Support Vector Machines in a channel current based signal analysis platform [4-9]. In this paper, methods for improving feature extraction are presented to improve both classification and to provide biologists and chemists with a better understanding of the physical properties of a given molecule.

Nanopore, Hidden Markov Models, Support Vector Machines, Emission Variance Amplification, Feature Extraction, Channel Current Cheminformatics

Introduction

A critical step in establishing a channel current-based signal analysis platform is a careful selection of features from the channel current blockade signal. A twist on the standard implementation of a HMM, emission inversion, that improves classification is presented. The addition of a feature representing spike density is also examined and shown to notably improve classification results. Other features that were studied included a full expansion of transition information as opposed to the compressed transition information used in the previous architecture [4]. Emission Variance Amplification is also introduced to the HMM in an effort to obtain accurate dwell time or level duration information for the observed levels of a given molecule. Finally, these expanded features introduce redundant, noisy information as well as diagnostic information into the current feature set and thus degrade classification performance. A hybrid Adaptive Boosting approach is used for feature selection to alleviate this problem.

Ultimately, classification results are the measure of success of the usefulness of a given feature or feature set. Emission inversion and the addition of a spike density feature are shown to noticeably improve performance and are folded into previously presented architecture. It is also shown that EVA greatly reduces computation complexity and makes analysis of levels that are not well defined possible, but an over-zealous use of tuning parameters can destroy kinetic information and thus render a channel current blockade signal useless. A new, efficient HMM-with-Duration is proposed as a solution [1-3]. Finally, although AdaBoost was not able to reproduce the best classification results obtained from a carefully selected feature set, AdaBoost is shown to be useful in several situations. Moreover, AdaBoost serves to validate the current, manually designed feature set.

Background

Nanopore Detector

The nanopore detector is the starting point of the channel current cheminformatics signal analysis architecture. That is, the nanopore detector generates the data analyzed in later stages of the architecture. A lipid bilayer supports the α-hemolysin channel, a protein heptamer formed by protein molecules secreted by *Staphylococcus aureus* that is used as the channel in the nanopore device due to its stable conformation (minimal gating) and its overall geometry (see Figure 1).

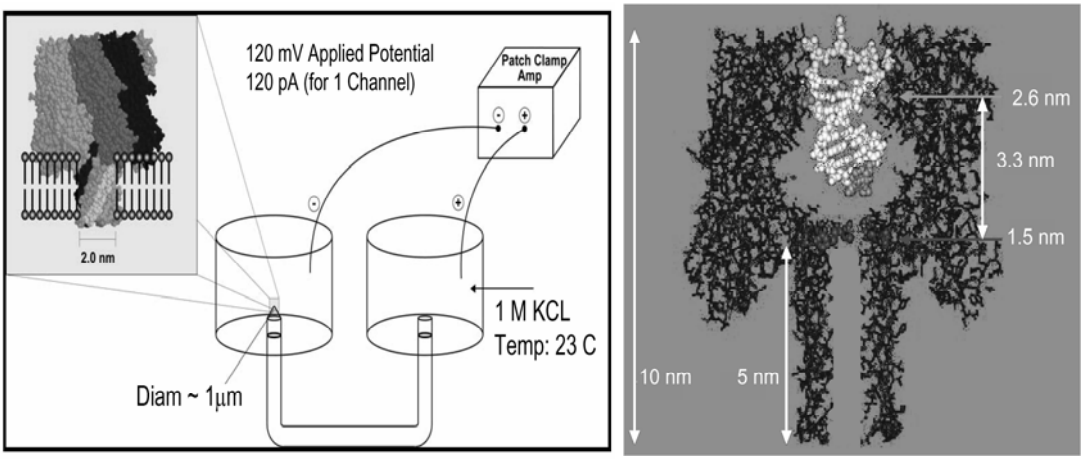


Figure 1. Left Panel: A lipid bilayer supports the alpha-hemolysin heptamer that creates a pore, or channel used to collect the data, as shown left. The channel is supported by an aperature, which allows the flow of ions between cis (here, left) and trans (here, right) wells. Right Panel: The assembled α-hemolysin pore shown to scale, with a captured dsDNA molecule. As shown, the double stranded form is too wide to pass through the pore, while a single strand may pass through.

This channel essentially collects the data. DNA and RNA interaction with the channel during translocation is non-negligible, but not strong enough for the molecule to get "stuck." Although dsDNA is too large to translocate, about ten base-pairs at one end can still be drawn into the large *cis*-side vestibule. This permits very sensitive experiments since the ends of "captured" dsDNA molecules can be observed for extensive periods of time to resolve features, allowing highly accurate classification of the captured end of dsDNA molecules [4-11]. In previous experiments, single molecules such as DNA have been examined in solution with nanometer-scale precision using nanopore blockade detection [4-9]. In early studies [9], it was found that complete base-pair dissociations of double stranded DNA to single stranded DNA could be observed for sufficiently short DNA hairpins. In later work [4-7], the nanopore detector was used to read the ends of double stranded DNA molecules and was operated as a chemical biosensor. In [5, 6, 10, 11], the nanopore detector was used to observe the conformational kinetics of the end regions of individual DNA hairpins.

Cheminformatics

The prototype channel current cheminformatics signal processing architecture "closes the loop" on the architecture previously presented in [4] (see Figure 2). The signal processing architecture is used to perform a preliminary test of pattern recognition informed (PRI) sampling control. As the nanopore detector generates data, a simplified time-domain Finite State Automaton (τFSA) shown in Figure 3 is used for signal acquisition (see [4, 12] for full model). Once the signal is acquired, it is passed on to a generic HMM that is used to characterize current blockades and extract features [1,2,4-7]. During this step, the parameters of a generic-HMM are estimated using Expectation Maximization (EM) to effectively de-

noise the signal [13]. After this stage, the extracted feature vector is passed on to an off-line-trained SVM. The classification result yielded by the SVM is then used to close the sampling control loop. In this paper additional techniques, improvements, and extensions to the machine learning techniques, primarily in feature extraction and feature selection are presented.

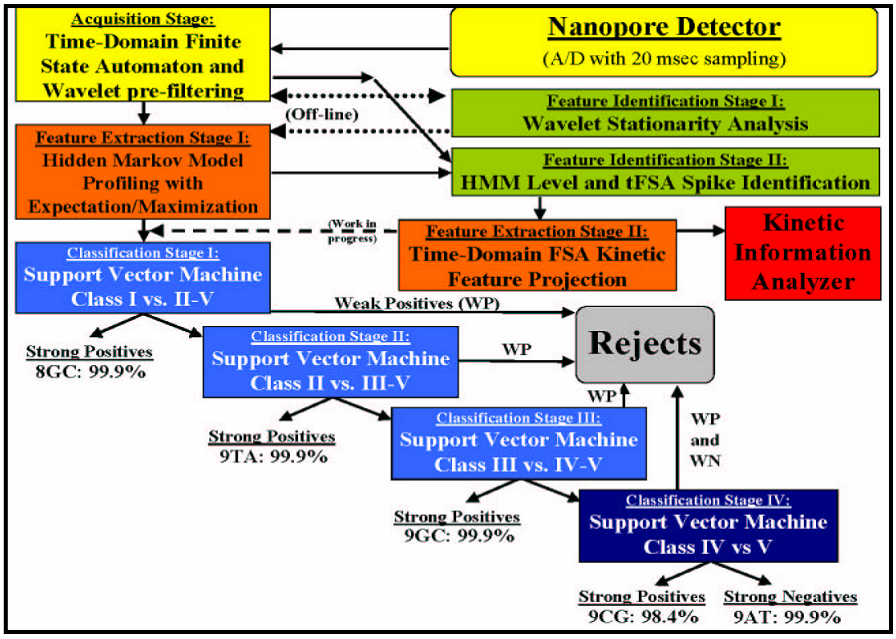
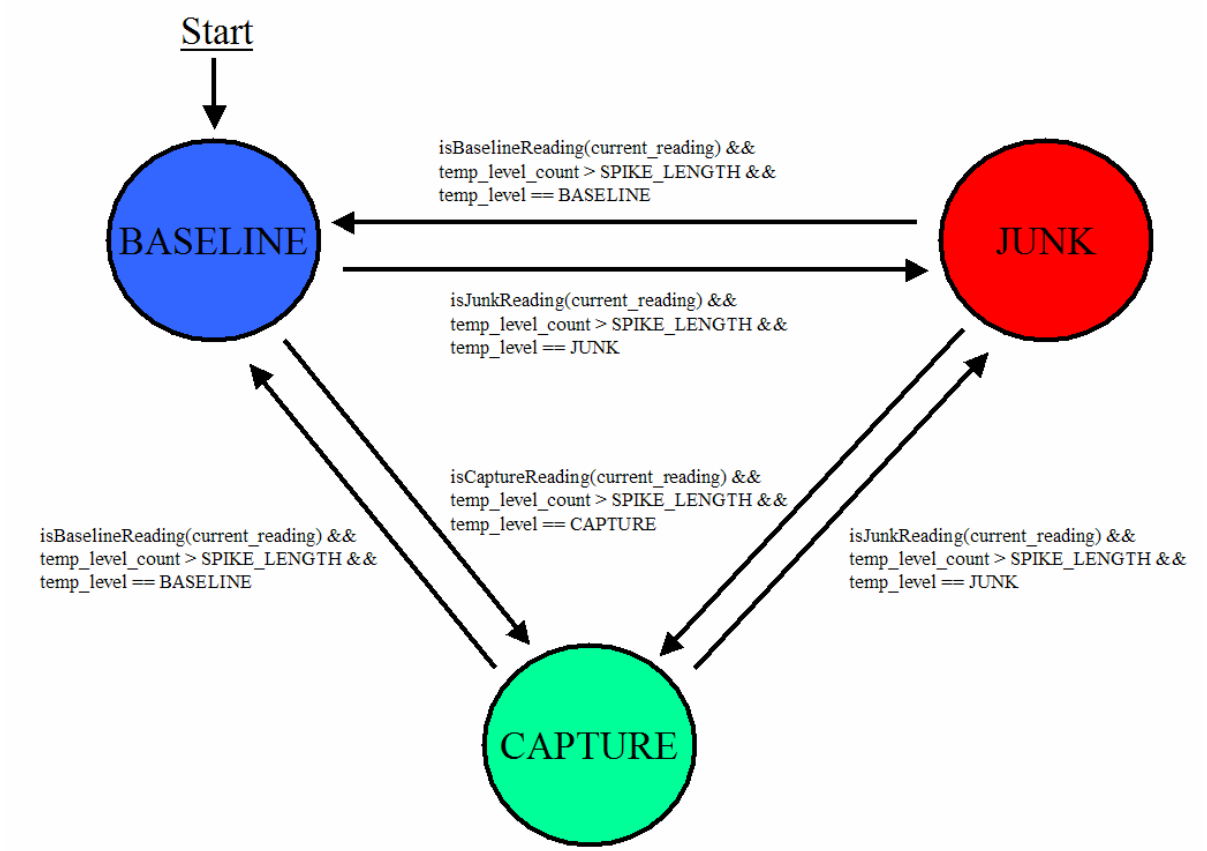


Figure 2. An overview of the channel current cheminformatics signal processing architecture.



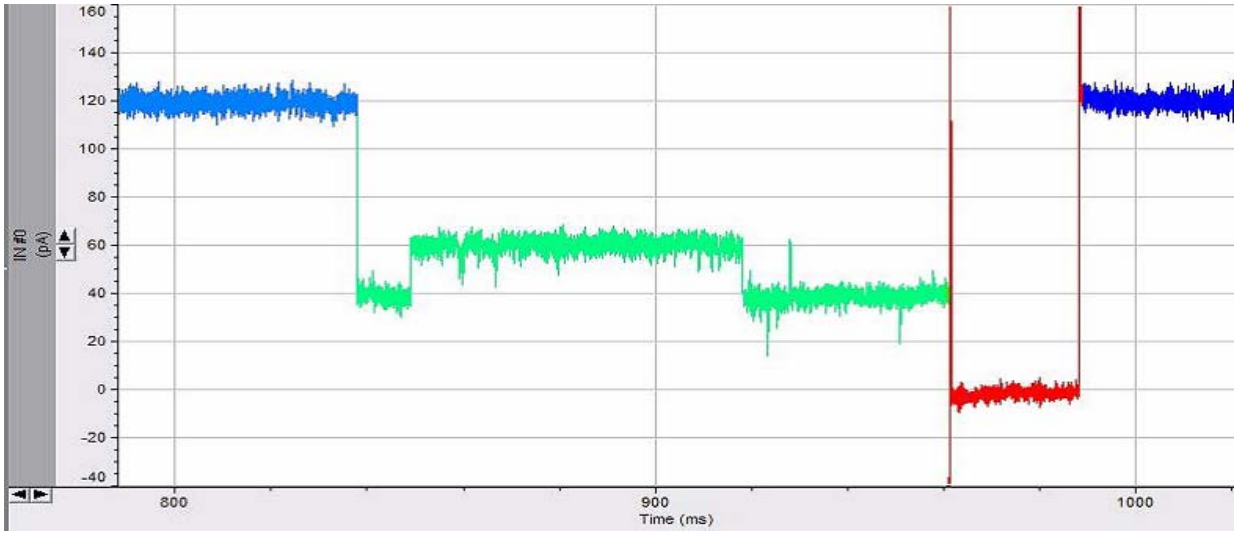


Figure 3. Top: A description of the FSA that is used to find captures in a channel current data file. Only transitions between states are shown. Staying in the current state does not require any updating of the state of the FSA. Transitioning to another state requires only the recording of that sample index if the capture state is entered or exit. Note that only the current reading of the current observation and the current level count are needed to determine the state of the current observation. The current reading is used to determine the level and the current level count is used to ensure an actual level and not noise in the channel. Bottom: A sample channel current blockade signal colored to correspond with the FSA in Figure 3. Note that the levels show significantly different properties, so a simple FSA will accurately find captures.

HMM Feature Extraction

A HMM is used to slightly de-noise and extract features from the acquired channel current signal. The HMM is implemented with fifty states. The only parameter necessary for determining a state is the current reading (which is given in picoamps) at a given point in the signal. This current reading is normalized to the baseline (the average current reading just

before a capture event occurred) of the signal and fit to a bin size of one. For example, an average baseline reading of 120pA and a current reading of 70pA corresponds to a normalized value of 58.33% baseline. Then, using a bin size of one, the value of 58 is used as the current state. For most of the data studied in these experiments, almost all capture events take place between 20- and 70% blockade. Thus, only fifty states are used in an effort to help ease computational complexity—computation time scales quadratically as input scales linearly. In the implementation of the HMM, the states are scaled with this observation in mind. In the previous example, our state of 58 would correspond to state 38. This process of scaling raw data to actual states is referred to as "quantization."

After the data is quantized, five rounds of Expectation Maximization are run to obtain accurate estimates of emission and transition probabilities. Initially, emissions for all states are set to a gaussian with mean L and unit variance. In addition, all transitions are equally likely. Expectation Maximization serves to obtain a more accurate measure of emissions and transitions based on the observed signal. A standard Viterbi algorithm is then run in order to de-noise the signal—that is, obtain the most likely path of states that created the observed signal. The process of finding the most likely path of states obtained by the Viterbi algorithm essentially reduces the noise in the channel current signal.

After the Viterbi algorithm is run, a 150-component feature vector is created for the given signal. Each feature vector consists of three distinct sets of information. The first 50 components come directly from the 50 previously described states of the HMM. These components are level occupation probabilities (a histogram view) for each state that are calculated after the Viterbit trace back algorithm yields a most likely path. The second set of 50 components is composed of the variances of the emission probabilities. The third and final set of 50 components is composed of a weighted sum of transition probabilities from the dominant levels of a given signal.

One refinement to the standard implementation of a HMM, presented here, involves the initial manipulation of the emission probabilities as they are entered in to the HMM. The emission probabilities are the main place where the observed data is brought into the HMM-EM algorithm and can be viewed conceptually as the probability of emitting a hidden or true state given an actual or observed state. By switching the roles of the true and actual states, it is believed that a type of entropy is introduced into the standard Viterbi dynamic programming table. While the exact theoretical underpinnings of this method are still being researched, it is clear that this "emission inversion" improves classification performance.

In addition to the 150-component feature vectors and the emission inversion technique already described, additional kinetic information can also be extracted. The effects of the addition of a spike density feature are explored, where a spike is defined as an anomalous, deep blockade of channel current from the lower level (lowest dominant level of channel current blockade) of a given signal.

Another variation on a standard HMM, Emission Variance Amplification is discussed. Here, the goal is to obtain dwell time information for the levels of a given molecule. From this information, the half-life, and thus, the stability of a given level can be determined. However, channel current data is noisy and building a Finite State Automaton to accurately model this noisy data can be difficult. Moreover, this model would not be easily re-usable for other channel current analysis without significant restructuring and re-tuning. Here, a HMM with EVA is used to reduce the gaussian noise bands around a given level while still strictly

retaining transitions between levels. This method was first introduced in [1] and is further explored here.

AdaBoost

Adaptive Boosting (AdaBoosting) is typically used for classification purposes. In general, AdaBoost is an iterative process that uses a collection of weak learners to create a strong classifier. Training data is given a weight, and at each iteration the weak learners are trained on this weighted data. Weights for these data points are then updated based on the error rate of the weak learner and whether a given data point was classified correctly or not. The consensus vote at each iteration is treated as a hypothesis, and weights are given to a hypothesis based on its accuracy. At the end of the iterative process, final classification is done using all hypotheses and their corresponding weights (see Figure 4). In this way, AdaBoost is able to use a set of weak learners to generate a strong classifier.

Input: $S = \langle (x_1, y_1), (x_2, y_2), ..., (x_N, y_N) \rangle$ where $x_i \in X$ and $y_i \in Y = \{-1, +1\}$
Initialization $D_1(i) = \frac{1}{N}$, for all $i = 1, ..., N$
for $t = 1$ *to* T **do**

1.Train weak learners with respect to the weighted sample set $\{S, D_t\}$ and obtain hypothesis $h_t : X \rightarrow Y$

2. Obtain the error rates ϵ_t of h_t over the distribution D_t such that

$$\epsilon_t = P_{i \sim D_t}[h_t(x_i) \neq y_i]$$

3. Set

$$\alpha_t = \frac{1}{2} ln(\frac{1 - \epsilon_t}{\epsilon_t})$$

4. Update the weights

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & h_t(x_i) \neq y_i \end{cases}$$

where Z_t is the normalizing factor so that D_{t+1} is a distribution

5. Break if $\epsilon_t = 0$ or $\epsilon_t \geq \frac{1}{2}$.

end
Output: $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

Algorithm 1: The AdaBoost algorithm

Figure 4. The traditional AdaBoost algorithm [18].

As a classification method, one of the main disadvantages of AdaBoost is that it is prone to over training. However, AdaBoost is a natural fit for feature selection. Here, over training is not a problem since AdaBoost finds diagnostic features and those features are passed on to a classifier that does not suffer from over training such as a SVM. For this function, a modified form of AdaBoost is introduced.

SVM Classification

Support Vector Machines (SVMs) are variational-calculus based methods that are constrained to have structural risk minimization (SRM) such that they provide noise tolerant solutions for pattern recognition [15,16]. Simply put, an SVM determines a hyperplane that optimally separates one class from another (see Figure 5). Once learned, the hyperplane allows data to be classified according to the region in which it resides.

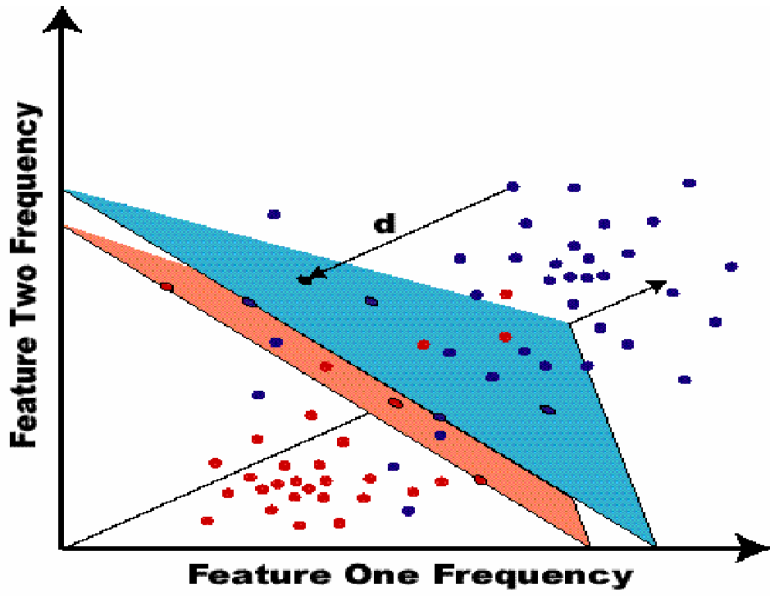


Figure 5. The hyperplane separability heuristic underlying the SVM classifier formulation, where the hyperplane is endowed with a thickness that is maximized (the SVM's structural risk minimization criterion).

The SVM approach encapsulates a significant amount of model-fitting information in its choice of kernel. In some sense, the SVM kernel provides a notion of distance to the decision hyperplane. Novel, information-theoretic, kernels were successfully employed for notably better performance over standard kernels in prior work[4,17].

Thus, SVMs are fast, easily trained, discriminators [15,16], for which strong discrimination is possible without the over-fitting complications common to neural net discriminators [15]. In these experiments, SVM classification performance is used as the benchmark for testing the validity of the various feature extraction permutations that are explored. This idea is a natural fit since one of the overarching goals of the nanopore detector is to be able to classify molecules based on their behavior in the channel. Furthermore, SVMs provide a natural confidence factor that can be leveraged when closing the sampling control loop.

Results

In what follows, results are described for the proposed extensions and improvements to existing methods in the feature extraction architecture. Improvements in feature extraction and types of features are discussed. Specifically, emission inversion, the addition of a spike density feature, and HMM with EVA is discussed. In addition, a new way of feature selection is shown. Here, the effects of using AdaBoost on a full set of transition probabilities versus a scheme for manually compressing transition probabilities are shown. The results presented refer heavily to data collected from single-molecule experiments on the nanopore detector.

Emission Inversion

Observed data is brought into the HMM/EM process chiefly through the emission probabilities. Through running the HMM in debug mode and observing the interactions of various components, an interesting twist on traditional emission probabilities was found—when the observed states and emitted states share the same alphabet the roles of observed states and emitted states can be reversed in order to improve classification performance.

Data used from these experiments were the 9bphp data shown in Figure 6. For all permutations of the binary classification problem with this data, three different feature sets were chosen to analyze the effect of emission inversion. The three sets selected for comparison were the manually designed 150-component feature vectors described in Background, the first set of 50 level occupation features from that 150-component set, and the second set of 50 variances on the emission probabilities from that 150-component set. The 9AT vs. 9TA, 9CG vs. 9TA, and 9GC vs. 9TA binary classification cases were selected to be shown here as they provide typical examples of the entire result set.

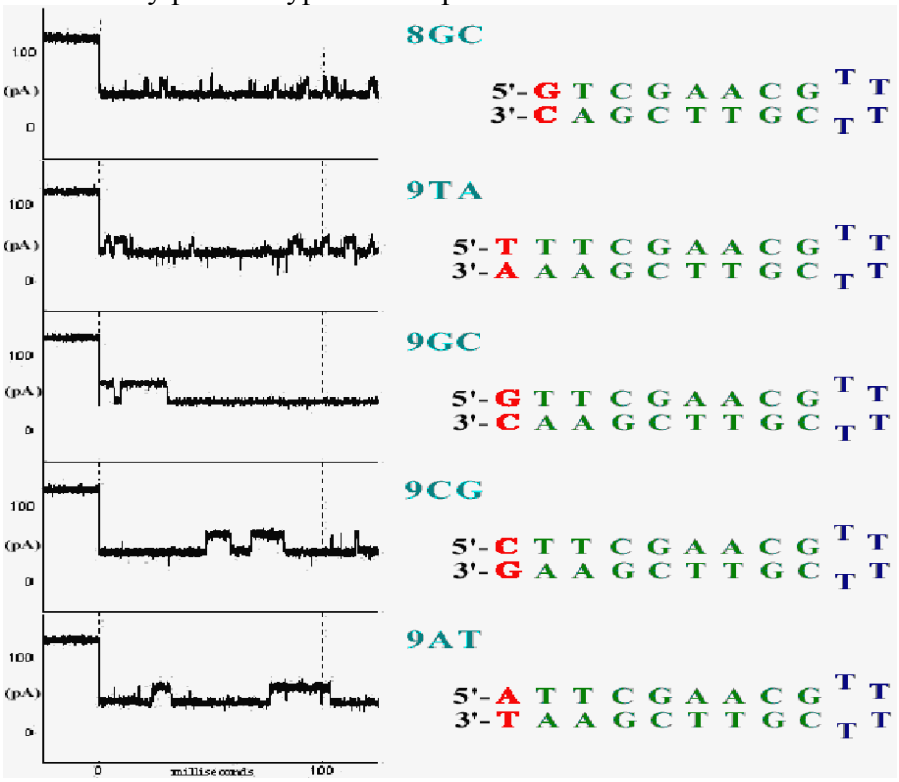


Figure 6. First 100 ms blockade patterns of four DNA hairpins, part of a test set of nine base-pair hairpins, with 4dT hairpin loops, that have been studied extensively, and an eight base-pair control. The nine base-pair molecules only differ in their terminal base-pairs, yet their channel current blockade signals, "signatures", are easily resolved [4].

Experimentally, this emission inversion works well with channel current data as shown in Figure 7. These figures show SVM classification performance for the various feature sets just described using both a standard HMM implementation and a HMM implemented with emission inversion as described here. The y-axis measures classification accuracy (sensitivity plus specificity) and the x-axis shows a tuning over the kernel parameter. The symmetric entropic kernel was used in this study as it has been shown to work well with channel current data in previous experiments [4]. The performance benefit is shown most notably in Figure 7c. In the case where the 150-component feature set was used, inverting the emissions yields a 5% peak increase in accuracy. This result is stable over a range of kernel parameter. For the case where the first 50 components were studied, a slight increase in classification performance as well as an increase in stability is observed. In the final case, a slight boost in classification performance is observed while a significant increase in stability is observed.

In nearly all cases studied, inverting the emissions provides a performance increase in accuracy, stability, or both accuracy and stability. For some molecules, this performance increase was more significant than others and in one case out of the ten permutations studied, performance was better using a standard HMM without emission inversion.

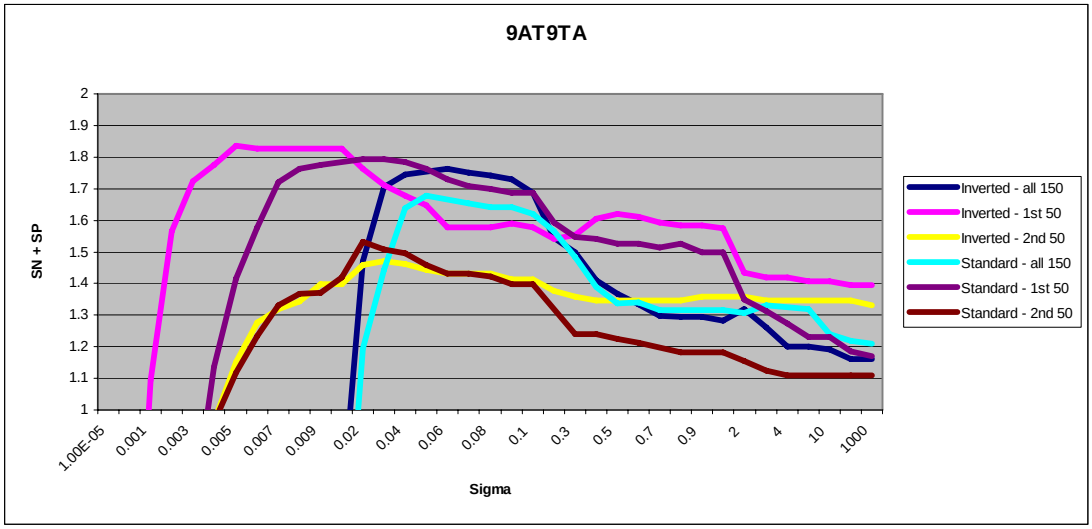


Figure 7a.

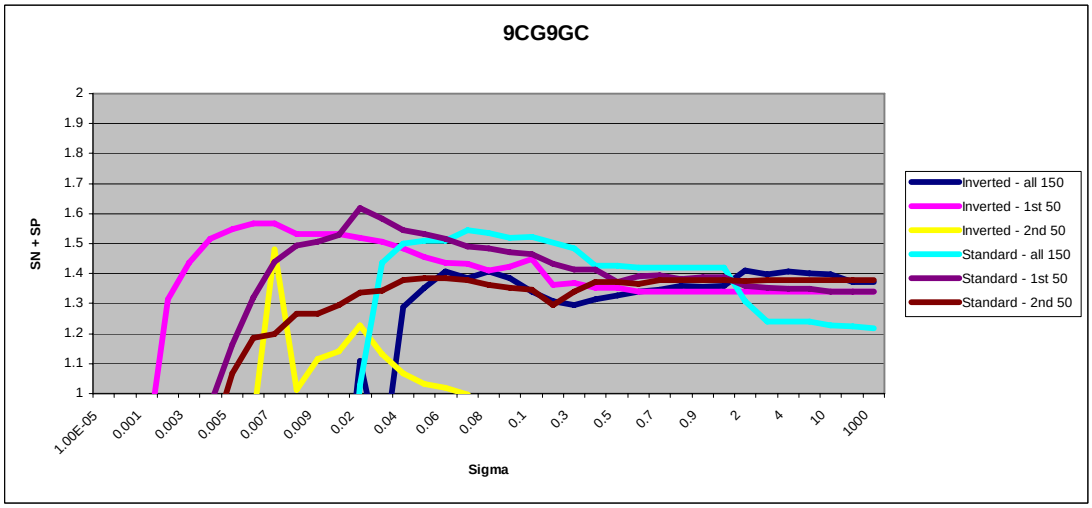


Figure 7b.

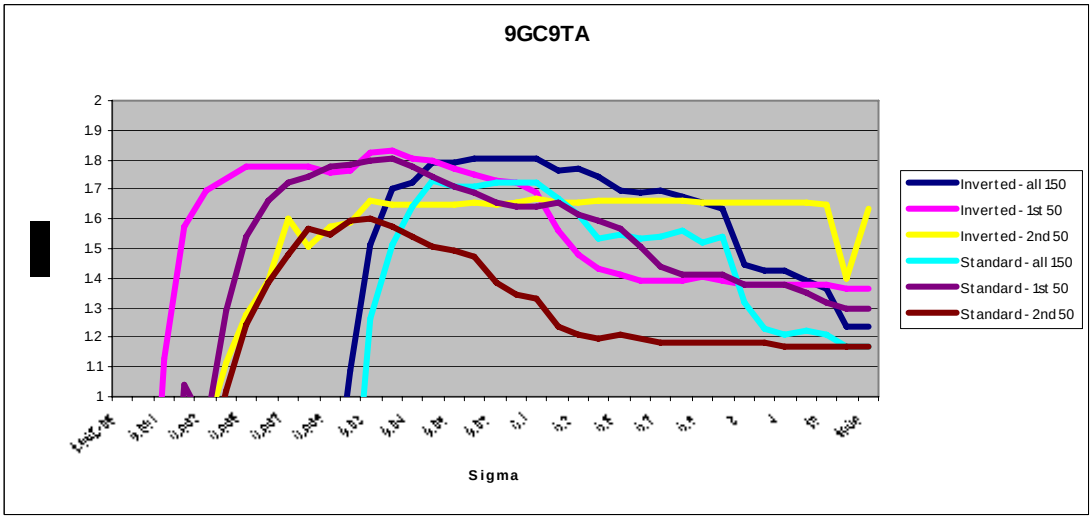


Figure 7c. SVM performance with different feature sets, for different binary classification data sets: (a) 9AT vs 9TA; (b) 9CG vs 9TA; and (c) 9GC vs 9TA. Throughout, the SVM shows that the feature set produced using the inverted emissions performs consistently better than the standard implementation of a HMM.

Spike Analysis

In addition to the level occupation probability, emission probability, and transition probability, the spike density from the lower level of a given molecule has been identified as a possibly significant feature. A spike event—an anomalous, deep blockade of channel current—from the lower level is conceptually seen as a fraying of the last few termini of a given molecule. Thus, a measure of spike density can yield information about the stability of the final few base pairings.

For this analysis, data obtained from collaborators at NASA/AMES was used. Here, the analysis is centered on two very similar 9GC molecules. On one of the molecules, the terminal guanine base was modified in an effort to simulate radiation damage. A blockade level histogram of the two signals (Figure 8) shows that there is high similarity between the blockades produced by the two molecules.

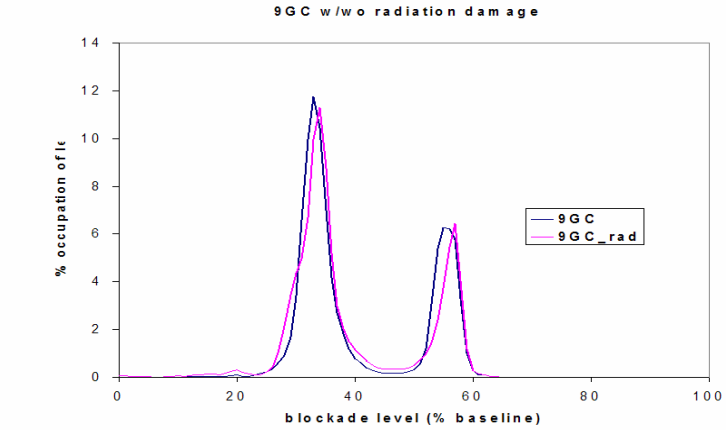


Figure 8. A blockade level histogram of 9GC and 9GC radiated data. Note how similar the two curves are – this will pose complications for mutual discrimination based solely on these features).

The spike detection method presented in [13] was used to identify spikes and extrapolate true spike counts shown in Figures 9 and 10.

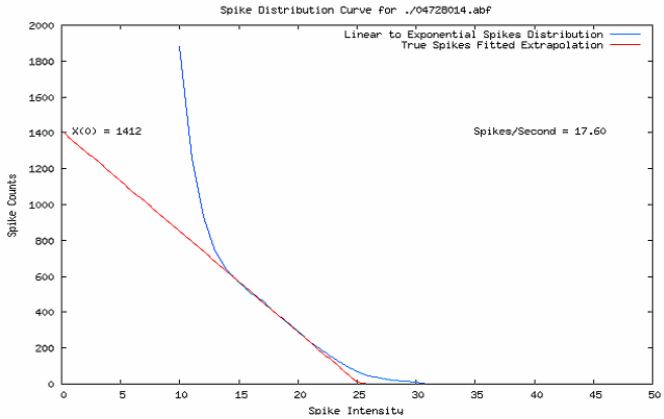


Figure 9. A plot used for spike density estimation. The blue curve represents actual spike counts observed versus a given cutoff. The red curve is drawn tangent to the observed curve. Thus, the true spike count is the reading as the tangent line crosses the x-axis. The molecule studied is a 9 base-pair hairpin that is the radiation damaged DNA model (a terminal guanine is oxolated) (see [10] for details), with terminal guanine unaltered in the "non-radiated" molecule. The spike count plots show increasing counts as spike cut-off thresholds are relaxed (to where eventually any downward deflection will be counted as a spike). Plots are automatically generated using gnuplot and automatically fit with extrapolations of their linear phases at the group's tools website. The extrapolations provide an estimate of "true" anomalous spike counts – counts associated with terminus fraying in the captured DNA hairpin (as shown in [8]). The radiated form of the molecule frayed 17.6 times on average (while in the LL state).

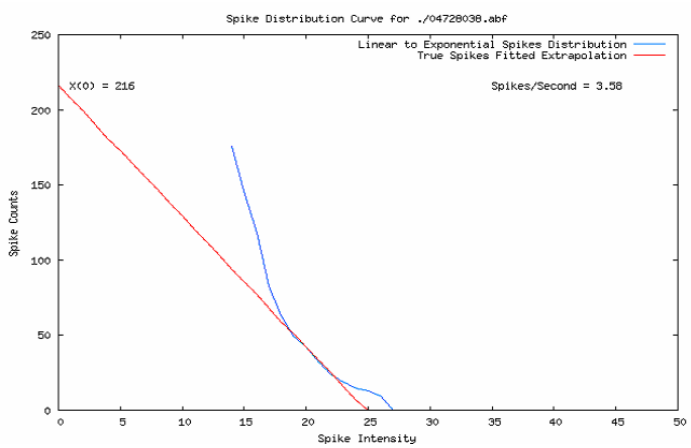


Figure 10. The non-radiated molecule only frayed 3.58 times a second, on average, while in its lower-level state.

Building on the efforts in [13], this spike density feature was used as a single feature and concatenated to the end of the 150-component feature vector (described in Background). The results of this analysis are shown in Figure 11 (similar to the description of the emission inversion results in the previous section). Incorporation of this spike feature for this data set leads to classification with approximately 5% greater accuracy over a wide range of tuning parameters. It is noteworthy that the addition of only one extra feature, the spike density feature, yields a significant performance increase.

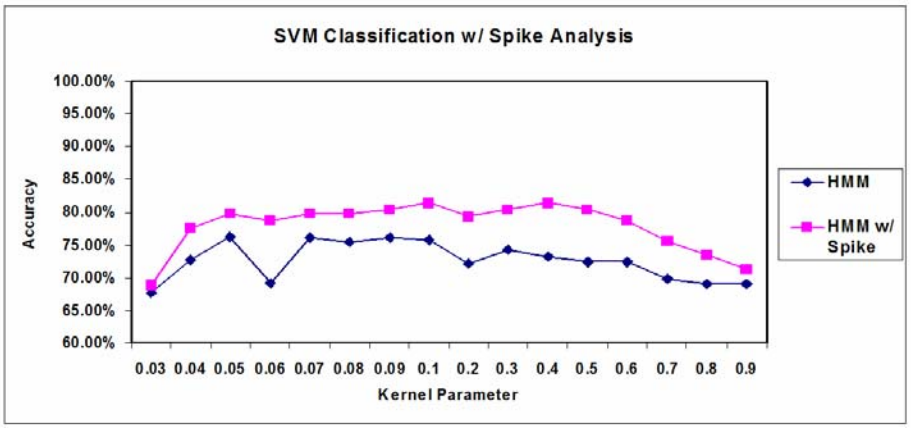


Figure 11. Example classification results with and without spike analysis. Note that adding a spike feature significantly improves classification accuracy over a wide range of kernel parameters.

Dwell Time Analysis

Another important feature of a channel current blockade signal is the duration of blockade levels. However, acquiring level duration information is a non-trivial task due to a significant gaussian noise band around blockade levels. The goal here is to use Emission Variance Amplification in the HMM with EM to drastically reduce noise in the signal while still retaining level transitions. By retaining the level transitions, the integrity of the kinetic information—level dwell times in this case—remain in tact.

Data used for this analysis was gathered from a simple study of DNA annealing with the nanopore detector and a Y-aptamer transduction platform. Results on blockade states observed for Y-aptamer overhang+complement binding study are shown in Figure 12.

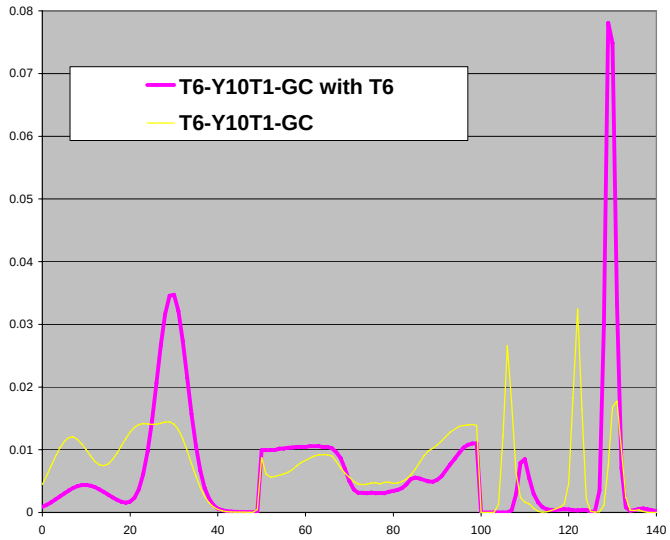


Figure 12. The 150-component feature vector profiles for the Y-aptamer that binds a 6A ssDNA, for signals before and after introduction of that six adenosine ssDNA.

Visually, the results of EVA can be seen in Figure 13. Note that as the variance is amplified from the original setting of 1, the noise band around a given level is reduced significantly. Moreover, even though many spike events are destroyed, transitions between dominant levels—and thus level dwell times—are strongly retained. Now, a trivial Finite State Automaton can extract dwell time information. This FSA only needs a current reading and a duration (in sample counts) to characterize any given level. Without EVA, a wide range of current cutoffs or even some more complex model would be needed to characterize a given level. But, using this simplified FSA, dwell time distributions for the studied data were easily obtained (see Figure 14). From these dwell time distributions, the half-life—and thus a measure of level stability—can be gathered. This half-life is an important kinetic characteristic for a biologist or chemist studying the properties of a molecule. Future work will evaluate whether half-lives of levels or even entire dwell time distributions can be useful in improving classification performance.

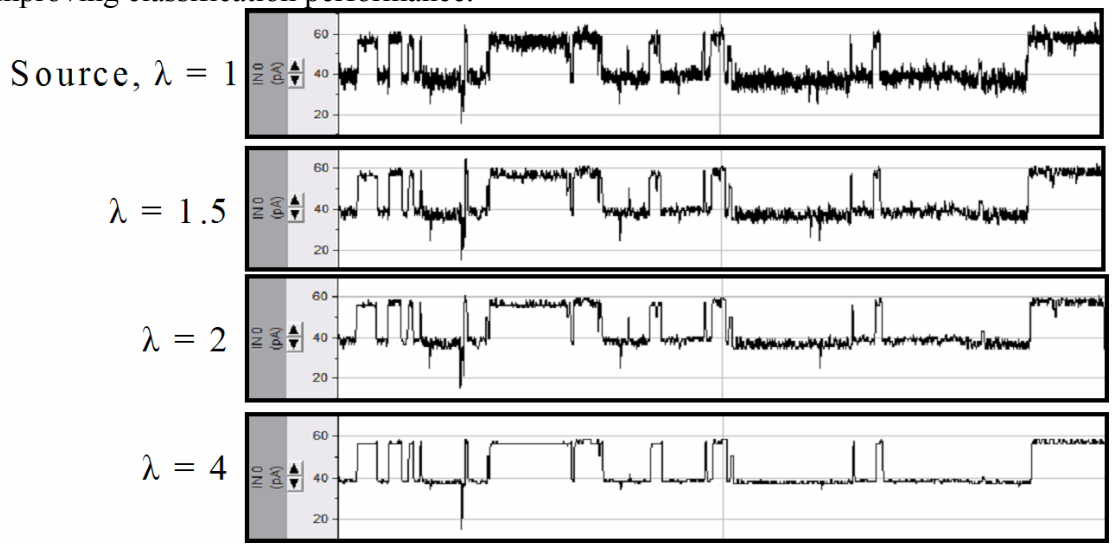


Figure 13. As the EVA factor increases, the gaussian noise surrounding the levels is reduced significantly, yet level transitions are strictly retained.

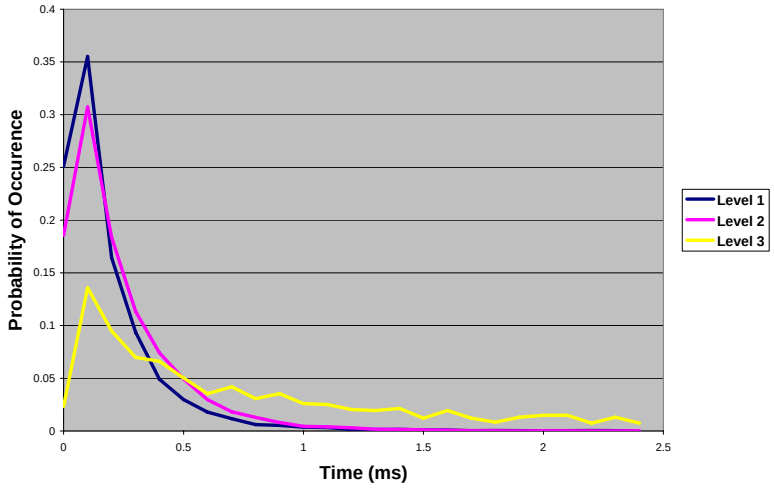


Figure 14. The dwell time distributions for the three dominant levels of the Y-aptamer (without 6A target). For further details and Results, see the work presented in [19].

Feature Selection

As has been shown in the spike analysis, careful selection of features plays a significant role in classification performance. However, adding non-characteristic or noisy features will hurt classification performance. In addition, recall from the discussion in Background that the last set of 50 components from the baseline 150-component feature vector are compressed transition probabilities. With a 50 state HMM, there would be 50*50 or 2500 possible transitions. However, a means of compression is necessary because many of these transitions are very unlikely and contribute noise to the feature vector. Without compression, classification performance suffers as a result, yet it is uncertain as to whether diagnostic information has been inadvertently discarded in the manual compression of the transition probabilities. An automated approach is desired to solve the issue of feature selection. Here, a hybrid AdaBoost approach is used as a "hands-off" means of feature selection.

The data studied for feature selection include the 9CG vs 9GC and 9GC vs 9TA binary classification problems from the 9bphp data used in the emission inversion analysis (Figure 6). The 9GC vs 9TA set was studied first. Since the 9GC vs 9TA case is one of the easier classification problems with this dataset, the 9CG vs 9GC case was also analyzed. This case is among the hardest binary classification problems in this dataset.

Figures 15, 16, and 17 show the results of this automated feature selection analysis (these figures have a similar description to the figures described in the Emission Inversion results section). Figure 11 shows the effects of AdaBoosting off of the full, uncompressed feature vectors. These feature vectors are comprised of the 50 blockade level components (same as from the 150-component set), the 50 variances on the emission probabilities (same as from the 150-component set), and the full 2500 transition probabilities. Using a SVM to classify all 2600 features shows a notable decrease in classification accuracy and a significant decrease in the stability of classification results. AdaBoost is used to select the top 100 diagnostic features. These 100 features are extracted from the full 2600-component set of features and passed on to the SVM for classification. In this case, classification outperforms both the full 2600-component set and the manually designed 150-component set. The curve denoted by "First 50" represents the first 50 blockade level probabilities. This set is the best performing manually designed set, and outperforms the AdaBoost selected feature set in both performance and stability. Figure 17 shows the results of AdaBoosting off of the manually

designed 150-component feature set in the case of the 9GC vs 9TA binary classification problem. There is a notable performance increase in classification accuracy and stability.

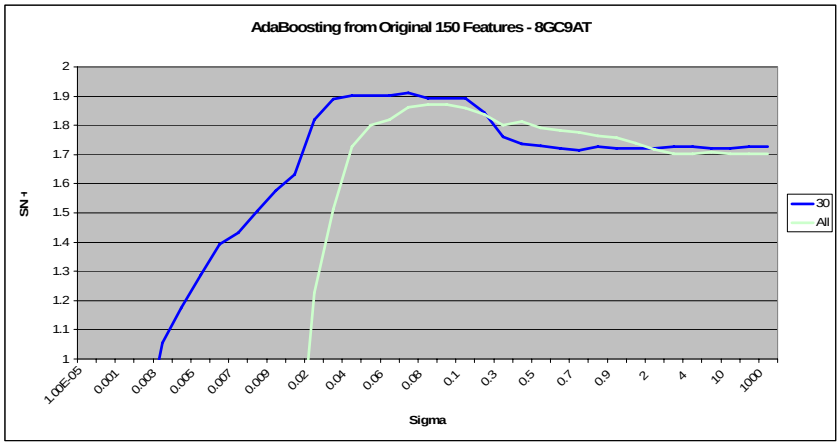


Figure 15. If Adaboost operates from the 150-component manual set, a reduced feature set of 30 is found to work best, and with notable improvement in kernel parameter stability in the region of interest.

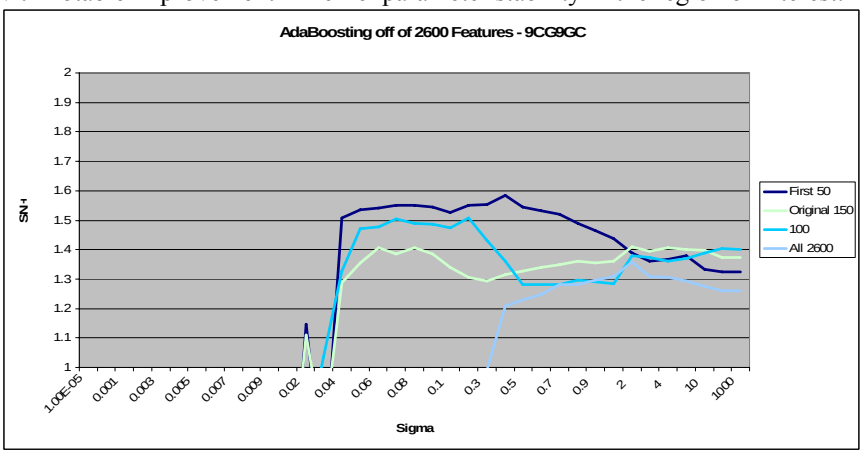


Figure 16. AdaBoosting to select 100 of the full set of 2600 features improves classification over just passing all 2600 components to the SVM. Bit the best performance is still obtained when working with the Adaboosting from the manual set.

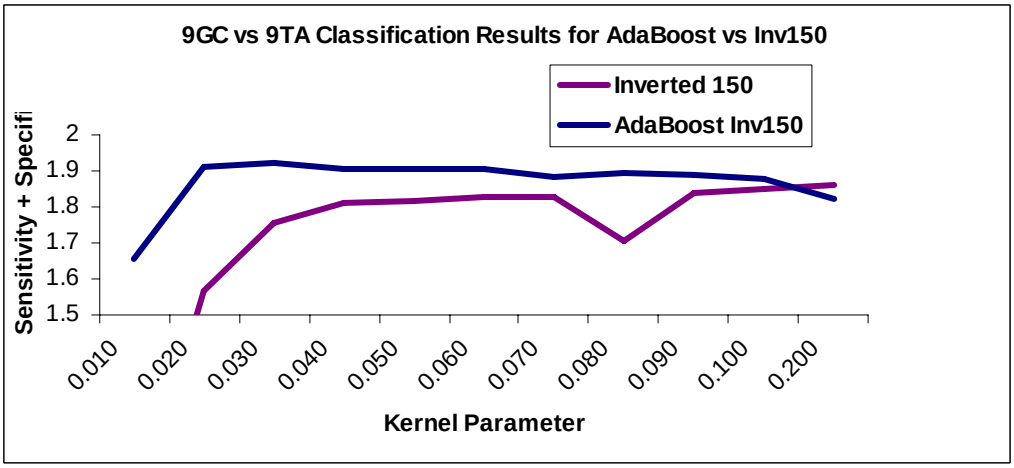


Figure 17. Classification improvement with Adaboost taking the best 50 from the Inverted-emission 150-component feature set. 95% accuracy is possible for discriminating 9GC from 9TA hairpins with no data dropped with use of Adaboost, without Adaboosting, the accuracy is approx. 91%. This demonstrates a significant robustness to what the SVM can "learn" in the presence of noise (some of the 2600 component have richer information, but even more are noise contributors). This also validates the effectiveness with which the 150 parameter compression was able to describe the two-state dominant blockade data found for the nine base-pair hairpin and other types of "toggler" blockades, as well as the utility of the inverted features.

Discussion

In what follows, the pros and cons of each proposed method presented in the Background and Results sections are discussed. In addition, proposed fixes and future work is discussed.

Emission Inversion

Admittedly, emission inversion is currently a tricky topic. Emission inversion denotes a changing of the roles of emitted states and observed states. The exact theoretical underpinnings of swapping these roles are not yet completely understood. In some sense, however, classification performance is the ultimate judge of the validity of a given method. As described in the Results section, SVM classification results do just that.

There are currently a couple of caveats. This emission inversion only works where the emitted and observed states share the same alphabet. In the current channel current blockade analysis platform, this restriction holds. Another caveat is that this method may be data dependent. Only channel current data has been studied and it is entirely possible that emission inversion does not generalize to other datasets. In this case, the AdaBoost feature selection presented in this paper may provide a simple fix. Simply create datasets that include extracted features from both a standard HMM implementation and a HMM implementation with emission inversion and let AdaBoost select the most diagnostic features in an automated way.

Spike Analysis

The results described above clearly show that spike density from the lower level is an important feature. Obtaining the spike density feature (described in Methods) is straightforward. However, adding this feature to the existing 150- or 2600-component feature sets currently requires tuning. Simply adding the spike density feature to an existing feature vector already containing 150 features will obscure the effect of the spike density feature almost completely. Thus, a weight must be added to this new feature. Should the weight be too heavy, though, the effect of the other features will be obscured. Currently, the weighting factor is tuned over in order to arrive at a weighting such that the spike density feature improves classification without obscuring the contribution of other features.

A few automated solutions are suggested for future work. One proposed solution is to simply add the un-weighted spike density feature to the existing feature vector and use AdaBoost to select the most diagnostic features. This approach will essentially create a weight for the spike density feature. That is, by removing many components that only add noise to a given feature vector, the remaining features are given more weight. Another solution that is currently being worked on is to fold the definition of a spike into the HMM. This solution requires a non-trivial amount of work as the entire definition of a state has to be entirely reworked. Moreover, the definition of a state must be considered carefully such that a state explosion (as seen in higher order HMMs) does not occur.

Dwell Time Analysis

Preprocessing channel current blockade data using a HMM with EVA significantly reduces the complexity of dwell time analysis. Within a reasonable range of values for EVA factor, the noise bands around levels are significantly reduced while level transitions are retained. However, if too large of an EVA factor is used then transitions can be destroyed and the channel current signal will be mangled beyond use. Although this problem is not

significant for a wide range of EVA factor, a HMM with Duration [1-3] will retain transitions and can eliminate this problem altogether.

Another aspect of the dwell time analysis that will be explored in future work is the effect of dwell time information on classification. Dwell time distributions for dominant levels should be characteristic for a given signal and thus improve classification performance. However, a significant amount of data is generally necessary to generate accurate dwell time distributions. In the current architecture, 300ms of channel current blockade are analyzed to create one feature vector. It is unclear as to whether 300ms will be enough data to overcome this limitation on sparseness of data. A longer signal trace could be analyzed, but computational complexity explodes quadratically as signal input increases linearly. Here, the use of a distributed HMM [14] could allow for the processing of enough data to provide accurate dwell time statistics while still meeting reasonable time constraints.

Feature Selection

Typically AdaBoost is used as a classification method. But due to the limitations discussed in the Background section, SVMs provide a much more robust means of classification for channel current data. However, AdaBoost is useful in feature selection. The weighting schemes in the AdaBoost algorithm are a natural fit for feature selection as the weights indicate which features are most diagnostic for a given classification problem.

AdaBoost does require some subtle tuning. As can be seen in the algorithm shown in Figure 4, AdaBoost does not have a natural end point. Unlike an SVM, AdaBoost does not converge on a solution. The number of iterations in the AdaBoosting algorithm must be tuned over in order to ensure accurate results. Another tuning parameter is the number of diagnostic features “D” to select from the original feature set “O”. Should D be chosen too small, diagnostic features existing in O will be excluded and classification performance will suffer. Should D be chosen too large, noisy features existing in O will exist in D and classification performance will suffer. In general, though, the choice of D does not present a great problem as SVMs are robust and can learn well in the presence of noise and non-diagnostic features. Experimentally it has been observed that it is more important that D not be chosen too small as opposed to too large (see Figure 18).

It is also important to note that automated feature selection using AdaBoosting was not able to reproduce results obtained from the “best-case” manually designed feature set (see Figure 16). Nonetheless, feature selection using AdaBoost is an important technique. It allows for the automated exploration of the effect of many different features and feature sets. In addition, AdaBoosted feature selection would be useful in problems where the definition of states do not lead to an easily designed manual set of features.

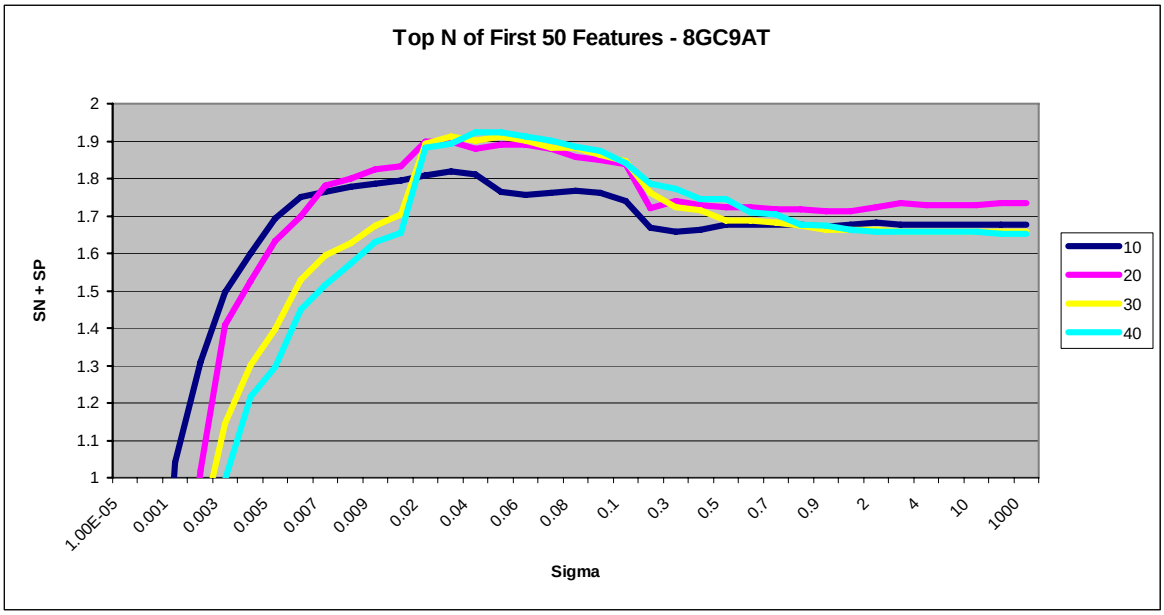


Figure 18. Notice that performance is relatively stable when the top 20, 30, or 40 performing features are selected. Performance drops nearly 5% when only the top 10 performing features are selected—selecting only 10 features does not capture all of the diagnostic information from the original feature set.

Conclusions

Several new techniques and improvements on existing techniques in the channel current signal analysis platform have been introduced. Emission inversion was introduced and was shown to be an improvement over the standard implementation of a HMM in regards to channel current data and final classification performance. Previous work on spike detection was folded into the current architecture. In addition, a new method for analyzing dwell times, Emission Variance Amplification was introduced to the HMM. Finally, a hybrid AdaBoost approach was introduced in an effort to improve the feature selection process. Not only are these techniques useful improvements for the current signal process architecture, but several techniques introduced here also provide means to move forward with future research as detailed in the Discussion section.

Methods

Emission Inversion

As previously discussed in the Background and Discussion sections, the main place where data is introduced into the HMM/EM algorithm is through the emission probabilities. In the HMM, emissions are defined as a multidimensional array and can be viewed conceptually as the probability of a hidden state emitting an observed state:

emission_probabilities[actual_state][observed_state].

A standard implementation of a HMM would be implemented in the following manner:

```
For (I = 0; I < NUM_STATES; I++) {
        Forward[0][I] = emission_probabilities[I][observed_data[0]] *
                Prior_probability[I];
}
```

The emission inversion implementation simply swaps the roles of the actual state and the observed state as follows:

```
For (I = 0; I < NUM_STATES; I++) {
        Forward[0][I] = emission_probabilities[observed_data[0]][I] *
                Prior_probability[I];
}
```

This simple inversion seems to introduce another information factor into the Viterbi algorithm and improves performance as discussed in the Results section.

Dwell Time Analysis

As mentioned in the Discussion section, a HMM with EVA is used to significantly reduce the gaussian noise band around levels. In a non-EVA approach, emission probabilities are initialized with a gaussian profile. The initialization is as follows:

emission_probabilities[i][k] = exp(-(k-i)*(k-i)/(2*variance))

where "i" and "k" are each a state with 0 <= i,k <= 49 in a 50 state system. To perform EVA, the variance is simply multiplied by a factor that essentially widens the gaussian distribution imposed on possible emissions, and the equation simply becomes

exp(-(k-i)*(k-i)/(2*variance*eva_factor)).

Essentially EVA boosts the variance of the distribution and yields the following effect: for states near a dominant level in the blockade signal, the transitions are highly favored to points nearer that dominant level. This is a simple statistical effect having to do with the fact that far more points of departure are seen in the direction of the nearby dominant level than in the opposite direction. When in the local gaussian tail of sample distribution around the dominant level, the effect of transitions towards the dominant level over those away from the dominant level can be very strong. In short, a given point is much more likely to transition towards the dominant level than away from it.

Feature Selection

As introduced in the Backgrounds and Discussion sections, AdaBoost is used in feature selection. In this hybrid implementation, weights are given to the weak learners as well as the training data. The key modifications here are to give each column of features in a train set a weak learner and to update each weak learner every iteration, not just updates the weights on the data. Conceptually, this idea can be seen in Figure 19. Training data can be viewed as a two dimensional array of feature components. $F_1 - F_j$ are individual feature vectors representing a single capture event. $E_1 - E_i$ are the experts or weak learners assigned to an individual component in feature space. In the implementation described in this paper, naïve bayes classifiers were used as weak learners.

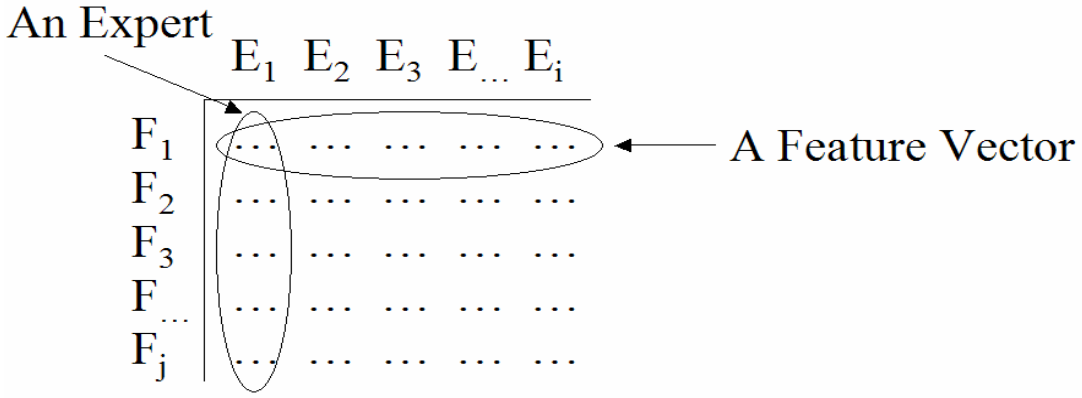


Figure 19. A representation of the adapted AdaBoost method used for feature selection. Each feature vector F_i has an associated weight as in standard AdaBoost implementations. For feature selection, each expert E_i is also given an associated weight.

For a given number of iterations T, the process is as follows:

Initialize weights on weak learners
Initialize weights on training data
For i, 1 .. T
    Train weak learners
    Update the weights for each weak learner – just like the hypothesis update in the standard AdaBoosting method
    Update the weights for each training point – just like the original AdaBoosting method
    Normalize the two weights.
    Break if the overall composite learner's error rate is 0% or 50%

In an example where there is a set of 150-component feature vectors, 150 weak learners would be created. As previously mentioned, each weak learner corresponds to a single component and classifies a given feature vector based solely on that one component. Then, weights for these weak learners are introduced. In each iteration of this modified AdaBoost process, weights for both the input data and the weak learners are updated. The weights for the input data are updated as in the standard AdaBoost implementation while weights on the individual weak learners are updated as if each were a complete hypothesis in the standard AdaBoost implementation (see Figure 4).

At the end of the iterative process, the weak learners with the highest weights, that is, the weak learners that represent the most diagnostic features, are selected and those features

are passed on to a SVM for classification. Thus, the benefits of both AdaBoost and SVMs are obtained.

Acknowledgments

Funding was derived from grants from NIH, NSF, LA Board of Regents, and NASA. Thanks to the Research Assistants that gathered the data: Eric Morales and Iftekhar Amin. A special thanks to Dr. Stephen Winters-Hilt, advisor.

References

1. Winters-Hilt S: Hidden Markov Model Variants and their Application. BMC Bioinformatics 2006, Sept. 26, 7 Suppl 2: S14.
2. Baribault, C and Winters-Hilt S. HMM-with-Duration Signal Analysis Tools, Real-Time Applications, and the Cheminformatics Web-interface. BMC Bioinformatics 2007 (submission).
3. Churbanov A, Baribault C, Winters-Hilt S. Duration learning for nanopore ionic flow blockade analysis. BMC Bioinformatics 2007 (submission).
4. Winters-Hilt S., W. Vercoutere, V.S. DeGuzman, D.W. Deamer, M. Akeson, and D. Haussler. 2003. **Highly Accurate Classification of Watson-Crick Basepairs on Termini of Single DNA Molecules.** Biophys. J. 84:967-976.
5. Winters-Hilt, S., **Nanopore detection using channel current cheminformatics**, *SPIE Second International Symposium on Fluctuations and Noise,* 25-28 May, 2004.
6. Winters-Hilt, S. and M. Akeson, **Nanopore cheminformatics**, DNA and Cell Biology, Oct. 2004.
7. Winters-Hilt, S. 2003. **Highly Accurate Real-Time Classification of Channel-Captured DNA Termini.** *Third International Conference on Unsolved Problems of Noise and Fluctuations in Physics, Biology, and High Technology,* pg 355-368.
8. Vercoutere, W., S. Winters-Hilt, V. S. DeGuzman, D. Deamer, S. Ridino, J. T. Rogers, H. E. Olsen, A. Marziali, and M. Akeson. 2003. **Discrimination Among Individual Watson-Crick Base-Pairs at the Termini of Single DNA Hairpin Molecules.** Nucl. Acids Res. 31, pg 1311-1318.
9. Vercoutere, W., S. Winters-Hilt, H. Olsen, D.W. Deamer, D. Haussler, and M. Akeson. 2001. **Rapid discrimination among individual DNA hairpin molecules at single-nucleotide resolution using an ion channel.** Nat. Biotechnol. 19(3):248-252
10. Winters-Hilt S., Landry M, Akeson M, Tanase M, Amin I, Coombs A, Morales E, Millet J, Baribault C, and Sendamangalam S. Cheminformatics Methods for Novel Nanopore analysis of HIV DNA termini. BMC Bioinformatics 2006, Sept. 26, 7 Suppl 2: S22.
11. Winters-Hilt S, Davis, A, Amin, I, and Morales E. The Nanopore Cheminformatics of Individual Transcription Factor Binding Site Interactions. BMC Bioinformatics 2007 (submission).
12. Cormen, T.H., C. E. Leiserson, and R. L. Rivest. 1989. **Introduction to Algorithms.** MIT-Press, Cambridge, USA.
13. A. Prabhakaran, Power Signal analysis of Channel Current Signal using HMM-EM and Time-domain FSA, UNO MS Thesis in CS, Dec. 2005.
14. Jiang, Z and Winters-Hilt S. Distributed HMM Processing and Data Absorption. BMC Bioinformatics 2007 (submission).
15. Vapnik, V. N. 1998. **The Nature of Statistical Learning Theory** (2nd ed.). Springer-Verlag, New York.
16. Burges, C.J.C**.** 1998. **A tutorial on support vector machines for pattern recognition.** Data Min. Knowl. Discov., 2. 121-67.
17. Winters-Hilt S, Yelundur A, McChesney C, Landry M: Support Vector Machine Implementations for Classification & Clustering. BMC Bioinformatics 2006, Sept. 26, 7 Suppl 2: S4.
18. Iqbal R, Landry M, Winters-Hilt S: DNA Molecule Classification Using Feature Primitives. BMC Bioinformatics 2006, 7 Suppl 2: S15.

19. Thomson K, Amin I, Morales E, and Winters-Hilt S. Preliminary Nanopore Cheminformatics Analysis of Aptamer-Target Binding Strength. BMC Bioinformatics 2007 (submission).

Vita

Matthew Landry was born in Metairie, Louisiana. He was valedictorian of the Archbishop Rummel High School class of 2001. He received his B.S. in Computer Science from the University of New Orleans in 2001 and graduated Summa Cum Laudae.