Web Access Portals to the Stochastic Sequential Analysis Protocol and Methods

Stephen Winters-Hilt[1,2] , Jorge Chao[1,*], and Joshua Morrison[1]

[1] Dept. of Computer Science, University of New Orleans, 2000 Lakeshore Drive, New Orleans, LA 70148, USA
[2] Meta Logos Incorporated, 6218 Waldo Dr., New Orleans, LA 70122, USA

*email: jorge.luis.chao@gmail.com

Abstract

We describe here web access portals to the stochastic sequential analysis (SSA) Protocol tools, with test data already available in the guest directory for immediate exploration. We use a collection of machine learning programs ranging from *ad hoc* acquisition methods, to HMM-based feature extraction methods, to SVM-based classification/clustering methods. The software is all developed "in-house" (mainly in C/Perl). Our core machine learning tools involve innovative, *distributed*, methods for training support vector machines (SVMs) and Hidden Markov Models (HMMs). With multiple processing passes, and use of finite state automata (FSAs), we are able to extract robust features and obtain very accurate classification results. The Machine Learning web-interfaces described here are for both machine learning experts and non-experts. For non-experts, default values are specified on the key parameters. The machine learning example applications are in bioinformatics and cheminformatics. The SSA Protocol web access allows access to distributed implementations for significant speedup, as will be described in the methods and results.

1 Introduction

We are developing web interfaces to put sophisticated machine-learning methods, the methods in the stochastic sequential analysis (SSA) Protocol, within reach of non-expert researchers for general-purpose classification, knowledge discovery, feature identification, and feature extraction. Wherever possible, we are also attempting to implement distributed processing capabilities to enable real-time and on-line operation.

Nanopore detector operations informed by methods used in the SSA Protocol have a significant boost in data acquisition rates (by orders of magnitudes, in some situations). Such a capability is being used to accelerate nanopore detector sampling operation, and many of the examples in the Methods involve nanopore detector data. The SSA Protocol also has application as the backbone of the signal processing methodology for a new type of carrier-wave based communication that is based on stochastic signals with stationary statistics. Distributed processing capabilities are established for all stages of the SSA Protocol, and the Mata Logos Servers make use of this to enable more specialized algorithmic refinements. Distributed signal processing refinements at the critical HMM bottleneck (in the real-time SSA processing) are shown in the Results.

Stochastic carrier wave (SCW) signal processing occurs in both natural and engineered situations. Whenever Nature is observed with a sequence of observations that have stationary statistics (associated with equilibrium and near-equilibrium flow situations, for example), then the basis for SCW signal processing arises. SCW also parallels all electrical engineering carrier-wave methodologies where periodic wave methods are used in some modulation scheme, thus the number of engineering applications is enormous. AM heterodyning, for example, can be replaced with stochastic carrier wave with pattern recognition informed (PRI) heterodyning. In a similar manner, also have phase modulation equivalence: the standard periodic carrier wave approach has a coherent phase reference, while SCW introduces a stochastic carrier wave with stationary statistics 'phase', where we have similar capabilities as with phased-locked loop (PLL), for example, where the phase tracking is done on SCW encoded information.

A brief review of a collection of generalized Hidden Markov models (HMMs), critical to the stochastic sequential analysis (SSA) Protocol, are described in what follows (see Methods and [1]), and a synergistic union of the methods is described that arrives at a new form of carrier-based communication, where the carrier is not periodic but is stochastic, typically with stationary statistics. HMM with binned duration, and meta-HMM algorithmic methods, are shown to enable practical stochastic carrier wave encoding/decoding, where stochastic carrier wave signal processing is encountered in a number of settings in science and nanotechnology. The SSA toolsets have been used for nanopore transduction detector (NTD) signal analysis (of engineered NTD 'Nanoscope' signals) as well as application in gene structure identification [2].

The web interfaces to SSA Protocol methods at the various stages are as follows:

- Time-Domain Finite State Automaton (τ-FSA)
  - -- Channel current blockade identification & acquisition

-- Anomalous Blockade-Spike Detector

- Hidden Markov Model (HMM) Methods
    - -- Gap-Interpolating Markov Model
    - -- Hash-Interpolating Markov Model
    - -- Novel Implementation of HMM-with-duration
    - -- Novel Prokaryotic and Eukaryotic Gene Structure Identification Tools
    - -- HMM/EM-based channel current feature extraction
    - -- HMM/EM EVA-projection for τ-FSA kinetic feature extraction
- Support Vector Machine (SVM) Methods
    - -- Interface to Binary SVM with novel kernel & algorithmic variants
    - -- Expert SVM Interface -- can modify the alpha-selection heuristic, etc.*
    - -- Interface to Multiclass SVM
    - -- SVM Clustering Interface

See www.cs.uno.edu/~winters & www.meta-logos.com for further postings on web interface availability

2 Background

The SSA Protocol entails methods for signal acquisition, feature extraction, classification, and clustering. General-user web-interfaces to these tools are described with the methods summarized in the Methods. Some of the Web-interface tools, and sample data that will be made available, pertains to nanopore detector channel current measurements. The Nanoscope users of the SSA Server will be in this latter category of channel current cheminformatics type signal processing (briefly described in the Methods). The SSA Server utility is discussed in Sec. 2.1. The NTD Kit Components are listed in Sec. 2.2.

2.1 SSA Server Users

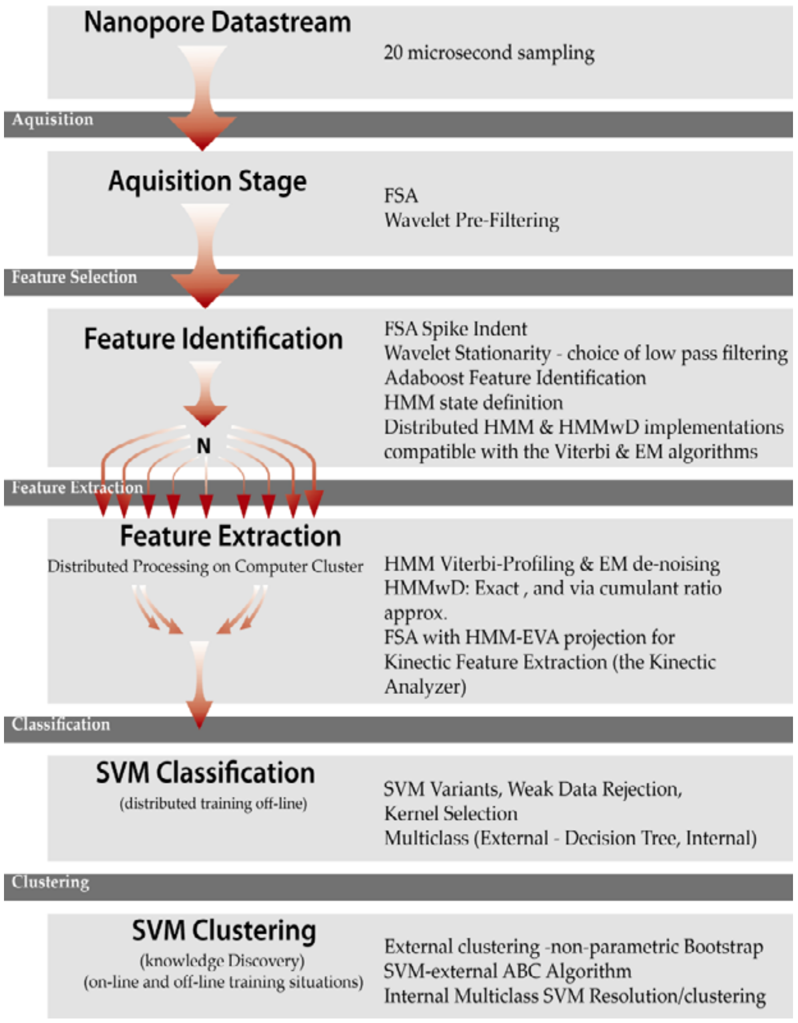
For Nanoscope Device or Nanoscope Kit users, a further refinement in the SSA Web-Server arrangement is to provide a streaming-data SSA-Server access. The latter setting offers Nanoscope calibration and troubleshooting capabilities, as well as access to distributed real-time speedup on core computational needs. Thus, the entire pipeline of signal processing needed, to go from raw streaming data to concise results, can be passed to the SSA-Server. Usually more efficient, and performed extensively in testing, is to separate the acquisition module (the tFSA) and bundle that with the device (or kit) and enable a bare-bones operational capability ‘locally’, then enhance with centralized maintenance of an SSA-Server algorithm toolset and data library. The latter is done in the real-time experiments shown in the Methods, where a network/device is set up as described. This same deployment is, thus, also sought in the Nanoscope Device or Nanoscope Kit development. What this means for the signal processing needs is that the latency of a large raw data transfer is largely eliminated by the local pre-processing to perform rudimentary signal acquisition, and then only passes acquired signal onto the network. For gene finding in mammals, for example, the gene regions amount to about 3% of the genome, so crude recognition of such would reduce acquisition and transmission of data by a factor of 33. The drop in bandwidth needs in channel current signal processing can be even more pronounced.

2.2 NTD KIT Components and Software

The Kit Package:
(1) NTD Kit Device Components and chemicals
(2) NTD Set-up Manual
(3) configured NTD Kit Computer with local SSA software and meta-logos SSA link
(4) buffer controls
(5) model systems/controls
(6) data analysis and data repository services
(7) Setup, calibration, and troubleshooting services

2.3 Channel Current Cheminformatics (CCC) Architecture

A protocol has been developed for the discovery, characterization, and classification of localizable, approximately-stationary, statistical signal structures in stochastic sequential data (see Methods), such as channel current data. Some of the CCC methods involved are shown in Fig. 1.



Nanopore Datastream
20 microsecond sampling
Aquisition
Aquisition Stage
FSA
Wavelet Pre-Filtering
Feature Selection
Feature Identification
FSA Spike Indent
Wavelet Stationarity - choice of low pass filtering
Adaboost Feature Identification
HMM state definition
Distributed HMM & HMMwD implementations
compatible with the Viterbi & EM algorithms
N
Feature Extraction
Feature Extraction
Distributed Processing on Computer Cluster
HMM Viterbi-Profiling & EM de-noising
HMMwD: Exact , and via cumulant ratio approx.
FSA with HMM-EVA projection for Kinectic Feature Extraction (the Kinectic Analyzer)
Classification
SVM Classification
(distributed training off-line)
SVM Variants, Weak Data Rejection,
Kernel Selection
Multiclass (External - Decision Tree, Internal)
Clustering
SVM Clustering
(knowledge Discovery)
(on-line and off-line training situations)
External clustering -non-parametric Bootstrap
SVM-external ABC Algorithm
Internal Multiclass SVM Resolution/clustering

Figure 1. Nanopore cheminformatics & data-flow control architecture. Aside from the modular design with the different machine learning methods shown (HMMs, SVMs, etc.), recent augmentations to this architecture for real-time processing include use of a networked server to link to the patch-clamp amplifier, and the 'real-time' pattern recognition informed signal processing architecture.

2.4 The SSA Toolset

For the general-user and the NTD-Kit user, the server interfaces for the SSA Server facilities are being designed to provide services ranging from web-interface based services to an active, streaming, experimental calibration/troubleshooting effort. Nanopore Device and Nanopore Kit users will have the signal acquisition module, the tFSA, bundled with their hardware (on a preconfigured computer, see Table 1). This allows for bare-bones functionality with the bundling as a stand-alone device. When there is need for more refined signal processing, such as when using the weakness recovery protocol to acquire signal when the FSA fails, the SSA Server will have the full panoply of methods available to 'get a lock' on the signal, and examine it further. The transfer of data at this juncture will also be efficiently pre-processed (e.g., the FSA acquisition will already done) and will thereby allow for much lower bandwidth and latency in the networked signal processing. Biology has invented such structure repeatedly, such as with the retinal pre-processing in the human eye.

KIT-user on-site Software Toolset	**General-user Web-Portal Toolset**
FSA	FSA, FSA-tuning
HMM*	HMM, HMMD, etc.
SVM*	generalized SVM
	Tuning Metaheuristics
	NTD Calibration Services
	NTD Troubleshooting Servies
	CCC Shared Reference Signal Library
(small data for HMM and SVM, no distributed processing)	(large datasets for HMM and SVM training, with use of distributed processing and speciality hardware such as GPUs)

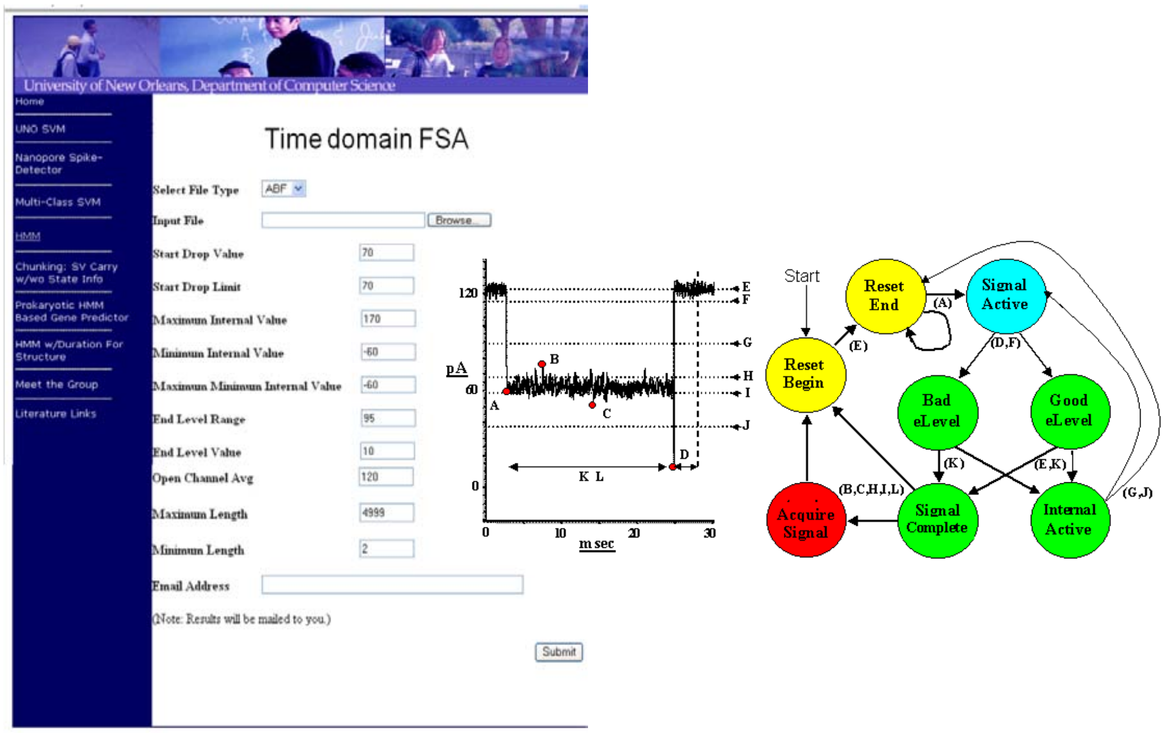
Table 1. Software Toolsets. *specialized enhancement packages to be made available from Meta Logos in future refinements to their Nanoscope effort.

3 SSA Protocol Methods and their Interfaces

Details of implementations possible with the Time-Domain Finite State Automaton (tFSA) are briefly described first (Sec 3.1), details on the weakness of HMMs are given in Sec. 3.2, the Generic HMM, introduced in [3-7,9,10], is described in Sec. 3.3. The clique-generalized HMM is described in Sec. 3.4; HMMD is described in Sec. 3.5; HMMBD in Sec. 3.6; adaptive HMM binning in Sec. 3.7; generalized SVMs in Sec. 3.8; External SVM-based clustering in Sec. 3.9; stochastic phase modulation (SPM) in Sec. 3.10; and the general SSA Protocol in Sec. 3.11.

3.1 The Time-Domain Finite State Automaton (tFSA)

The time-domain FSA is shown in Fig. 2, Right, where an eight-state automaton is shown with a six base-pair DNA hairpin channel blockade signal (from [9]). The Web interface to the acquisition interface for the automaton involves parameters commonly encounted in channel current blockade experiements, such as the blockade signal's drop in current value at onset of blockade, e.g., the "start drop value". This is an excellent acquisition parameter in the case of molecules that are engineered for channel capture and modulation in the NTD experiments).



Time domain FSA

Select File Type	ABF
Input File	
Start Drop Value	70
Start Drop Limit	70
Maximum Internal Value	170
Minimum Internal Value	-60
Maximum Minimum Internal Value	-60
End Level Range	95
End Level Value	10
Open Channel Avg	120
Maximum Length	4999
Minimum Length	2
Email Address	

(Note: Results will be mailed to you.)

Fig. 2. The Time-domain FSA. The web-interface comes pre-loaded with standard values for those not expert. Descriptions involving prior application of the methods [7], however, make their use by non-experts straightforward.

3.2 The Hidden Markov Model

Hidden Markov models are an amazing tool at the nexus where Bayesian probability and Markov models meet dynamic programming. To properly define/choose the HMM model in a machine learning context, however, further generalization is usually required. This is because the 'bare-bones' HMM description has critical weaknesses in most applications, which are summarized below. Fortunately, these weaknesses can be addressed, and in computationally efficient ways, see Fig. 3, with further details in the Methods.

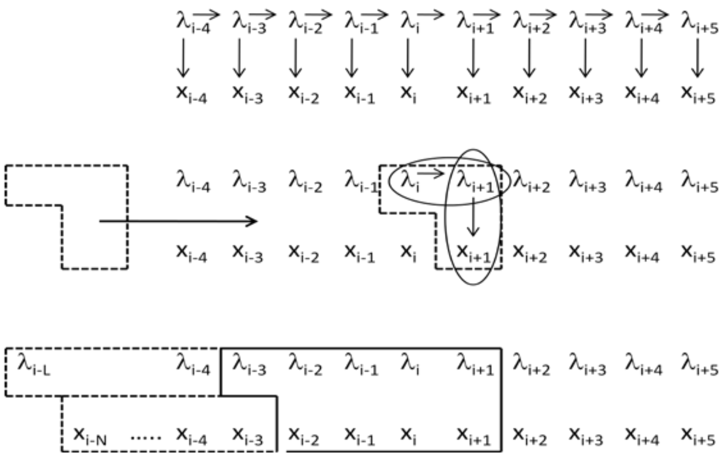

λi-4 λi-3 λi-2 λi-1 λi λi+1 λi+2 λi+3 λi+4 λi+5
xi-4 xi-3 xi-2 xi-1 xi xi+1 xi+2 xi+3 xi+4 xi+5
λi-L
xi-N .....

Fig. 3. Comparison of standard HMM and the clique-generalized HMM. The upper graphical model is for the standard HMM and shows the 'emission' observation sequence x_i, and the associated hidden label sequence λ_i, and the arrows denote the conditional probability approximations used in the model (for the transition and emission probabilities). Focusing at the level of the core joint-probability construct at instant 'i' in the middle graph, the standard HMM is a subset of the joint probability construct $P(\lambda_i, \lambda_{i+1}, x_{i+1})$. The generalized-clique HMM is shown in the graphical model at the bottom for one particular clique generalization. The model can be exact on emission positionally, then extend via zone dependence and use of gIMM interpolation. The model can be exact to higher order in state (referred to as footprint states', see [2]), and also extends modeling to have HMM with duration modeling. When doing the latter, zone-dependent and position dependent modeling can be incorporated via reference to the duration in the model, and can be directly incorporated into a generalized Viterbi algorithm (and other generalized HMM algorithms), as well as any other side-information of interest [3].

A brief list of the typical weaknesses encountered with the standard HMM:

(1) Standard HMMs are at low Markov order in transitions (first) and in emissions (zeroth), and transitions are decoupled from emissions, which can miss critical structure in the model (e.g., state transition probabilities that are strongly sequence dependent). This weakness is eliminated if we generalize to the largest state-emission clique possible, fully interpolated on the data set, with use of a minimal state-length constraint to obtain an efficient implementation (see Fig. 3).

The generalized clique HMM (Fig. 3) begins by enlarging the primitive hidden states associated with the individual base labels (as exon, intron, or junk) to substrings of primitive hidden states or footprint states. There is a key constraint, however, to keep the scaling of footprint states linear with footprint size: the footprint states are constrained to have self-transitions with a minimal length such that a footprint, and the mostly overlapping 'next' footprint, together can only have one transition between states of different type. The emissions are likewise expanded to higher order in the fundamental joint probability that is the basis of the generalized-clique, or 'meta-State', HMM. Further details on the meta-state HMM [2] are in the Methods.

(2) Need Method for directly incorporating side-information into the dynamic programming table based optimizations (used in the Viterbi and Baum-Welch algorithms, etc.). This is solved in [3], where an HMM is 'bootstrapped' into a HMM-with-duration, see Methods for brief description.

(3) Standard HMMs don't properly model self-transition durations, imposing a 'best-fit' geometric distribution on self-transition duration distributions instead. This weakness is eliminated if we generalize to a HMM-with-duration (HMMD) formalism, where direct modeling on self-transition duration distributions is incorporated (see Fig. 3). Standard HMMD methods are computationally expensive, however, when compared to Standard HMM. This weakeness can be

addressed, without loss of generality, via use of HMM with binned duration (HMMBD) representations. Further details on HMMBD [4] are in the Methods.

(4) Standard HMM and HMMD have latency bottleneck if full table computation is used on a lengthy data sequence, so need method for distributed processing‘chunking’ with overlaps sufficient for recovery. This is demonstrated in the Results.

(5) Need Method for HMM Feature Extraction Selection, Compression, and Fusion. A modified form of Adaboost is used for this purpose, see [7].

(6) Need Multitrack (Holographic) Generalization. In particular, need to show that hidden constraints can significantly limit model complexity, as seen in the clique generalization with application in gene-finding in [2], allowing significant scaling in multiple hidden-track (holographic) model complexity. In the Results we show the preliminary statistical support to justify the Two-track HMM alternative-splice gene-finder model.

(7) Need Method for Standardized HMM application to power signal data, and this is described in the Methods.

(8) Need Method for Standardized HMM usage: the SSA Protocol is described in the Methods.

All of the HMM generalizations and feature extraction methods discussed in what follows can be optimized for speed with binned durations and through dynamic (“null”) binning, distributed table-chunking, and GPU-usage. This allows the limiting speed constraint on the core HMMBD component in the SSA protocol to be eliminated.

3.3 The Generic HMM

An HMM that is designed to generate a particular signal need only have a few states and transitions. In reverse, this HMM ‘template’ can be used to detect signal with matching statistics. An HMM that is meant to generate a large family of signals, on the other hand, needs to have more states and associated transitions. The ‘Generic’ HMM or ‘grayscale’ template HMM is an example of this in the case of the channel current analysis applications in [9] and in many of the examples in this paper.

3.3.1 Viterbi Path

In the Viterbi algorithm, a recursive variable is defined: $v_{kn} = v_k(n) = v_k(b_n)$ = *“the most probable path ending in state λ_n=k with observation b_n”*. The recursive definition of $v_k(n)$ is then: $v_l(n+1) = e_l(b_{n+1}) \max_k [v_k(n) a_{kl}]$. From which the optimal path information is recovered according to the (recursive) trace-back:

$\Lambda^* = \text{argmax}_{\Lambda} P(B, \Lambda) = (\lambda^*_0, \ldots, \lambda^*_{L-1})$; $\lambda^*_n|_{\lambda^*_{n+1}=l} = \text{argmax}_k [v_k(n) a_{kl}]$, and where $\lambda^*_{L-1} = \text{argmax}_k [v_k(L-1)]$, for length L sequence.

The recursive algorithm for the most likely state path given an observed sequence (the Viterbi algorithm) is expressed in terms of v_{ki} (the probability of the most probable path that ends with observation $b_n = i$, and state $\lambda_n=k$). The recursive relation is lifted directly from the underlying probability definition: $v_{ki} = \max_n\{e_{ki}a_{nk}v_{n(i-1)}\}$, where the $\max_n\{\ldots\}$ operation returns the maximum value of the argument over different values of index n, and the boundary condition on the recursion is $v_{k0} = e_{k0}p_k$. The emission probabilities are the main place where the data is brought into the HMM-EM algorithm. An inversion on the emission probability is possible when the states and emissions share the same alphabet of states/quantized-emissions (details in Sec. 2.3.6). The Viterbi path labelings are, thus, recursively defined by $p(\lambda_i|\lambda_{(i+1)}=n) = \mathrm{argmax}_k\{v_{ki}a_{kn}\}$. The evaluation of sequence probability (and its Viterbi labeling) take the emission and transition probabilities as a given. Estimates on those emission and transition probabilities themselves can be obtained by an Expectation/Maximization (EM) algorithm that is known as the Baum-Welch algorithm in this context.

3.3.2 pMM/SVM

For start-of-coding recognition, for example, one can create a profile Markov model (pMM) based log-likelihood ratio (LLR) classifier given by $\log[P_{start}/P_{non\text{-}start}] = \Sigma_i \log[P_{start}(x_i=b_i)/P_{non\text{-}start}(x_i=b_i)]$. Rather than a classification built on the sum of the independent log odds ratios, however, the sum of components could be replaced with a vectorization of components:

$$\Sigma_i \log[P_{start}(x_i=b_i)/P_{non\text{-}start}(x_i=b_i)] \dashrightarrow \{\ldots., \log[P_{start}(x_i=b_i)/P_{non\text{-}start}(x_i=b_i)], \ldots.\}$$

These can be viewed as feature vectors (f.v.'s), and can be classified by use of an SVM. The SVM partially recovers linkages lost with whatever order of Markov model dependency that is imposed. For the 0th order MM in the example, the positional probabilities are approximated as entirely independent -- which is far from accurate. The SVM approach can recover statistical linkages between components in the f.v.'s in the SVM training process.

There are generalizations for the MM sensor and its SVM f.v. implementation, and all are compatible with the SVM f.v. classification profiling. Markov Profiling with component-sum to component feature-vector mapping for SVM/MM profiling, thus, encompasses use of MMs, IMMs, gIMMs, hIMMs, and ghIMMs [10,11], and SVM usage via "vectorization" to SVM/MM, SVM/IMM, SVM/gIMM classification profiling (with use of the SVM's confidence parameter).

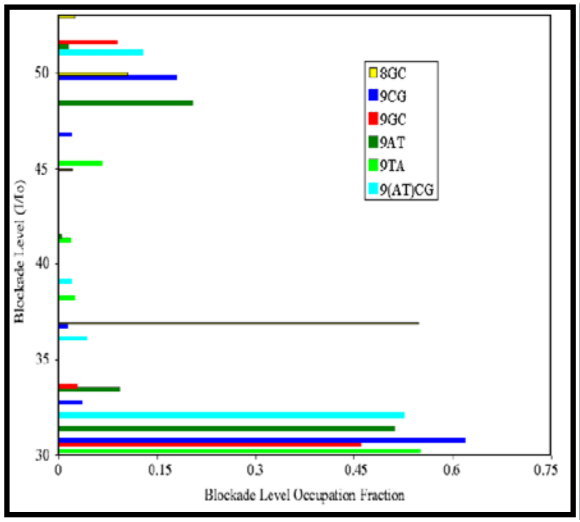
3.3.3 Feature Extraction via EVA projection (for use with power signal analysis)

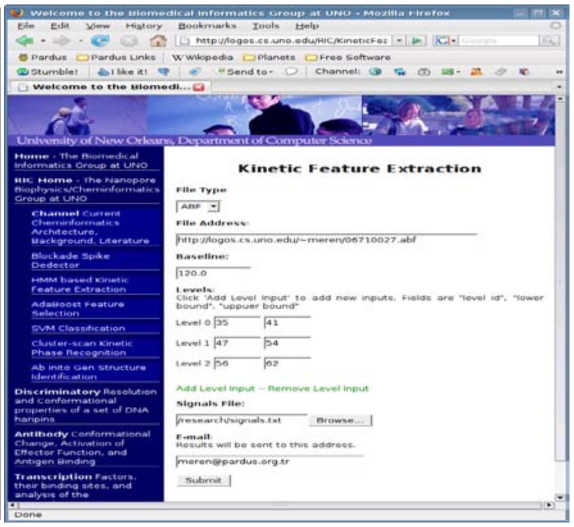
Emission variance amplification (EVA) projection is used in the SSA Protocol to go from a power signal (or anything sampled from a continuum domain of possibilities) to a sparser, projected 'EVA state', representation of the data. Quantization on the sparser representation can then provide a discrete representation. Once all states are discrete, higher order structure (or encoding) can be extracted by use of the meta-HMM generalization described in Sec. 3.2, and other methods.

In the CCC analysis in [9], we have an HMM with emissions probabilities parameterized by Gaussian distributions: emission_probabilities[i][k] = exp(-(k-i)*(k-i)/(2*variance)), where “i” and “k” are each a state where 0 <= i,k <= 49 in a 50 state system. To perform EVA in this setting, the variance is simply multiplied by a factor that essentially widens the gaussian distribution parameterized to best fit the emissions, and the equation simply becomes exp(-(k-i)*(k-i)/(2*variance*eva_factor)). For a sizable range of this parameter, HMM with EVA will remove the noise from the power signal while *strictly* maintaining the timing of the state transitions.

After EVA-projection, a simple FSA can easily extract level duration information. Each level is identified by a simple threshold of blockade readings, typically one or two percent of baseline. When EVA boosts the variance of the distribution, 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, thereby arriving at a “focusing” on the levels, while preserving level transitions. In this respect, other distributional parameterizations could be used than Gaussian, but Gaussian is a good starting point (a mixture of a sufficient number of distributions will arrive at a Gaussian distribution overall, an expression of the law of large numbers).

We have developed web interfaces for HMM based feature extraction (see Fig. 4). tFSA produces the signal files with the specified baseline. We load the signal files onto the KFE with the specified baseline and different blockade levels. The analysis results are emailed to the email address provided by the user.




Welcome to the Biomedical Informatics Group at UNO - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://logos.cs.uno.edu/RIC/KineticFe
Pardus
Pardus Links
Wikipedia
Planets
Free Software
Stumble!
I like it!
Send to
Channel:
Welcome to the Biomedi...
University of New Orleans, Department of Computer Science
Home - The Biomedical Informatics Group at UNO
RIC Home - The Nanopore Biophysics/Cheminformatics Group at UNO
Channel Current Cheminformatics Architecture, Background, Literature
Blockade Spike Dedector
HMM based Kinetic Feature Extraction
AdaBoost Feature Selection
SVM Classification
Cluster-scan Kinetic Phase Recognition
Ab inito Gen Structure Identification
Discriminatory Resolution and Conformational properties of a set of DNA hairpins
Antibody Conformational Change, Activation of Effector Function, and Antigen Binding
Transcription Factors, their binding sites, and analysis of the
Add Level Input -- Remove Level Input
Browse...
Submit
Done

Kinetic Feature Extraction

File Type

ABF

File Address:

http://logos.cs.uno.edu/~meren/06710027.abf

Baseline:

120.0

Levels:
Click 'Add Level Input' to add new inputs. Fields are "level id", "lower bound", "uppuer bound"

Level 0	35	41
Level 1	47	54
Level 2	56	62

Signals File:

/research/signals.txt

E-mail:
Results will be sent to this address.

meren@pardus.org.tr

Fig. 4. HMM/EM Viterbi-path level occupation feature extraction after strong EVA projection, and associated kinetic feature extraction interface used in further processing. For the CCC data shown, HMM/EM EVA (Emission Variance Amplification) Projection is used for simplified tFSA Kinetic Feature Extraction. The EVA projection shown here entails an HMM/EM filter

consisting of EM (5 loops) with the emission probabilities parameterized as gaussians, and then "projected" via a replacement gaussian with variance boosted by the factor indicated. The Plot generated makes it easy to then determine the level windows for the EVA-projected signals, for an FSA-scan type kinetic feature extraction with the interface shown on the right.

3.3.4 Feature Extraction via Data Absorption (a.k.a. Emission Inversion)

A new form of "inverted" data injection is possible during HMM training when the states and quantized emission values share the same alphabet. This is typically the case in the CCC power signal analysis examples given here. Results from channel current signal classification consistently show approximately 5% improvement in accuracy (sensitivity + specificity) with the aforementioned data inversion upon SVM classification (and this holds true over wide ranges of SVM kernel parameters and collections of feature sets). Transition & "absorption" statistical profiles are thought to work better than standard transition & emission profiles, in generalized classification performance, due to regularization with an effective SRM (structural risk minimization [11]) constraint, via optimization with an added term that depends on the relative entropy between state prior probabilities and emission posterior probabilities.

3.3.5 Modified AdaBoost for Feature selection and Data fusion

AdaBoost [7] can learn a sequence of weak classifiers and then boosts them by a linear combination into a single strong classifier. As a classification method, one of the main disadvantages of AdaBoost is that it is prone to overtraining. However, AdaBoost is a natural fit for feature selection. Here, overtraining is not a problem, as AdaBoost is only used to finds diagnostic features and those features are then passed on to a classifier that does not suffer from overtraining (such as an SVM). HMM features, and other features (from neural net, wavelet, or spike profiling, etc.), can be fused and selected via use of the Modified Adaboost selection algorithm [7].

3.4 The Meta-HMM – a clique-generalized HMM

The traditional HMM assumes that a 1st order Markov property holds among the states and that each observable depends only on the corresponding state and not any other observable. The meta-HMM entails a maximally-interpolated departure from that convention (limited according to the size of the training dataset) in an attempt to leverage anomalous statistical information in the neighborhood of non-self state transitions. The regions of anomalous statistics are often highly structured, having consensus sequences that strongly depart from the strong independence assumptions of the 1st order HMM. The existence of such consensus sequences suggests that we adopt an observation model that has a higher order Markov property with respect to the observations. Furthermore, since the consensus sequences vary by the type of transition, this observational Markov order should be allowed to vary depending on the state.

The gap and hash interpolating Markov Models (gIMM and hIMM) [10] can be directly incorporated into meta-HMMBD gene-finding models as a further enhancement to the underlying Markov models, since they are already known to extract additional information that may prove useful, particularly in the zone-dependent emission regions (denoted 'zde's as in [10]) where promoters and other gapped motifs might exist. Promoters and transcription factor binding sites often have lengthy overall gapped motif structure, and with the hash-interpolated Markov models it is also possible to capture the conserved higher order sequence information in the zde sample space. The hIMM and gIMM methods, thus, will not only strengthen the gene structure recognition, but can also provide the initial indications of anomalous motif structure in the regions identified by a gene-finder (in a post-genomic phase of the analysis) [10].

Use of the meta-HMM formalism resolves complications due to heavy-tail duration distributions and weak contrast. This is a new HMM modeling capability. The form of the clique factorization in [2] also has LLR terms such as $P(\widetilde{w}_n|ie)/P(\widetilde{w}_n|ii)$ that allow for a simple switch from internal scalar-based state discriminant to a vector-based feature, allowing for a similar substitution of a discriminant based on an SVM as demonstrated for splice sites in [5] and described in the pMM/SVM sub-section. These alternate representations do not introduce any significant increase in computational time complexity.

3.5 Hidden Semi-Markov model and HMM-with-duration

In the standard HMM, when a state i is entered, that state is occupied for a period of time, via self-transitions, until transiting to another state j. If the state interval is given as d, the standard HMM description of the probability distribution on state intervals is implicitly given by:

$$p_i(d) = a_{ii}^{d-1}(1-a_{ii})$$

where a_{ii} is self-transition probability of state i. As mentioned previously, this geometric distribution is inappropriate in many cases. The standard HMMD replaces the equation above with a $\boldsymbol{p_i(d)}$ that models the real duration distribution of state i. In this way explicit knowledge about the duration of states is incorporated into the HMM. When entered, state i will have a duration of d according to its duration density $\boldsymbol{p_i(d)}$; it then transits to another state j according to the state transition probability $\boldsymbol{a_{ij}}$ (self-transitions, $\boldsymbol{a_{ii}}$, are not permitted in this formalism). It is easy to see that the HMMD will turn into a HMM if $\boldsymbol{p_i(d)}$ is set to the geometric distribution shown above. The first HMMD formulation was studied by Ferguson [12]. A detailed HMMD description was later given by [13]. There have been many efforts to improve the computational efficiency of the HMMD formulation given its fundamental utility in many endeavors in science and engineering. Notable amongst these are the variable transition HMM methods for implementing the Viterbi algorithm introduced in [14], and the hidden semi-Markov model (HSMM) implementations of the forward-backward algorithm [15].

In [3] it is shown how to ‘lift’ side information that is associated with a region, or transition between regions, by ‘piggybacking’ that side information along with the duration side information. We use, as example, HMM incorporation of duration itself as the guide in what follows. In doing so, we arrive at a hidden semi-Markov model formalism for a HMMD. An equivalent formulation of the HSMM was introduced in [14] for the Viterbi algorithm and in [15] for Baum-Welch. In these derivations, however, the maximum-interval constraint is still present (comparisons of these methods were subsequently detailed in [16]). Other HMM generalizations include Factorial HMMs [17] and hierarchical HMMs [18]. For the latter, inference computations scaled as $O(T^3)$ in the original description, and havesince been improved to O(T) by [19].

3.6 HMMD with binned duration

The intuition guiding the HMMBD approach is that the standard HMM already does the desired duration modeling when the distribution modeled is geometric, suggesting that, with sufficient effort, a self-tuning explicit HMMD might be possible to achieve HMMD modeling capabilities at HMM computational complexity in an adaptive context.

The duration distribution of state i consists of rapidly changing probability regions (with small change in duration) and slowly changing probability regions. In the standard HMMD all regions share an equal computation resource (represented as D substates of a given state) -- this can be very inefficient in practice. In this section, we describe a way to recover computational resources, during the training process, from the slowly changing probability regions. As a result, the computation complexity can be reduced to O(TN^2+TND*), where D* is the number of "bins" used to represent the final, coarse-grained, probability distribution. A "bin" of a state is a group of substates with consecutive duration. For example, ***f(i, d)***, ***f(i, d+1)***, ...***f (i, d+δd)*** can be grouped into one bin. The bin size is a measure of the granularity of the evolving length distribution approximation. A fine-granularity is retained in the active regions, perhaps with only one length state per bin, while a coarse-granularity is adopted in weakly changing regions, with possibly hundreds of length states per bin. An important generalization to the exact, standard, length-truncated, HMMD is suggested for handling long duration state intervals – a "tail bin". Such a bin is strongly indicated for good modeling on certain important distributions, such as the long-tailed distributions often found in nature, the exon and intron interval distributions found in gene-structure modeling in particular. Starting from the above binning idea, for substates in the same bin, a reasonable approximation is applied:

$$\sum_{d'=d}^{d+\delta_d} f_t(i,d')\theta(b_t,i,d') \; = \; \theta\left(b_t,i,\bar{d}\right)\sum_{d'=d}^{d+\delta_d} f_t(i,d'),$$

where $\bar{d}$ is the duration representative for all substates in this bin.

3.7 Adaptive null-state binning for O(TN) computation

At each column the HMM Viterbi algorithm must look to the past column entries as it populates the table from left to right, thus leading to an O(TN^2) computation. If we establish an adaptive binning capability, reminiscent of what was done with the HMMBD method, then we can keep track of lists with respect to each state that correspond to prior column transitions to that state. If we, in particular, track those Viterbi most-probable-paths that arrive at our state cell with probability below some cutoff (with respect to the other probabilities arriving at that cell), we can ignore transitions from such cells in later column computations. What results is an initial O(tN^2) ($t \ll T$) computation to learn the state lists for above cut-off transitions (suppose K on average), followed by the main body of the O(TNK) computation (with $K \ll N$).

A method is also possible comprising use of a "fastViterbi" process where O(TN^2) $\rightarrow$ O(TmN) via learned, local, max-path ordering in a given column of the Viterbi computation for the highest 'm' values. Subsequent columns first only examine the top 'm' max-paths and if their ordering is retained, and their total probability advanced sufficiently, then the other states remain 'frozen-out' with a large grouping (binning) on the probabilities on those states used to maintain their probability information (and correct normalization summing) when going forward column-by column, with reset to full column evaluation on the individual state level when the m values fall out of their initially identified ordering.

A method is possible comprising use of a fastViterbi with null-binning process where $O(TN^2) \rightarrow O(Tmn) \rightarrow O(T)$ via learned global and local aspects of the data as indicated above. This approach offers significant utility as a purely HMM-based alignment algorithm that may outperform BLAST with comparable time complexity.

3.8 Generalized Support Vector Machines

Support Vector Machines are supervised learning methods used for classification and regression. They perform classification by optimally separating two sets of data with a maximally "thick" hyperplane (a simultaneous optimization in separability and thickness, see Fig. 5). They learn from training sets of labeled examples (the information that makes it supervised learning). Once trained, they assign un-labeled data a label, along with an associated confidence parameter in the implied classification.

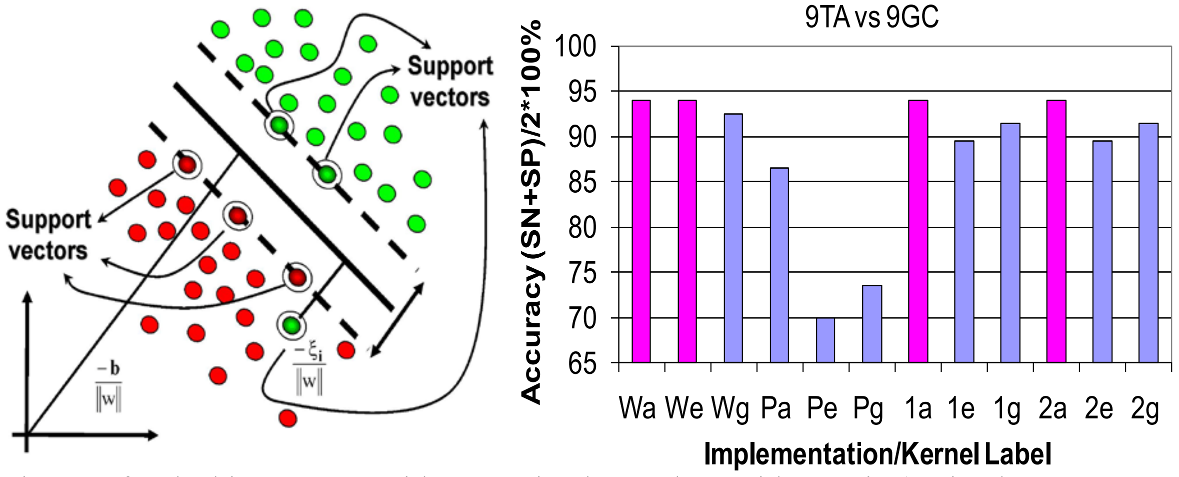
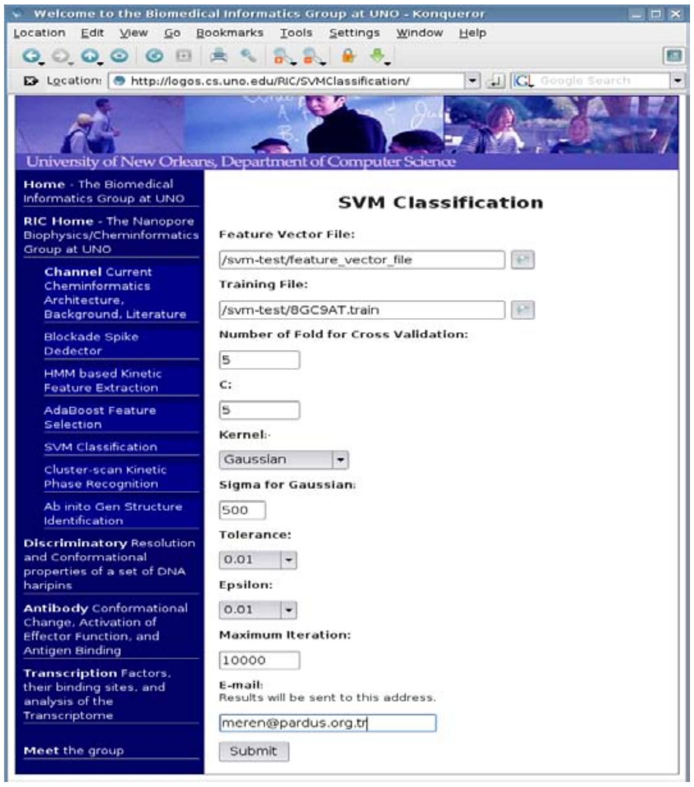


Fig 5, Left. The binary SVM with separating hyperplane with margin (region between dotted lines). The 'support vectors' that support the margin boundaries (and define them) are circled, as is the one outlier or misclassification, or non-separable data instance, that is shown. In the Right Panel are shown comparative results [25] involving the different kernels ('a' denotes absdiff, 'e' denotes entropic, 'g' denotes Gaussian, 'W' denotes use of the W-H SMO variant, 'P' denotes the Platt SMO variant, '1' denotes the first Keerthi variant, and '2' denotes the second Keerthi variant (see [25] for details). In the η-management implementation have: Platt SMO: η=2*K12-K11-K22;W-H SMO: η=2*K12-2; If (η>=0) {η=-1;}; and underflow handling and other implementations differ as well (see [25]).

We have developed a web interface that links to our Support Vector Machines implementation, as shown in Fig. 6. This will enable users to upload their training data and use SVM to classify the feature vectors according to the training data with a variety of kernels such as Gaussian, Absdiff, Sentropic, Dot etc. The report of the analysis will be emailed to the user at the given address.

Fig. 6. The SVM Web interface.


Welcome to the Biomedical Informatics Group at UNO - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: http://logos.cs.uno.edu/RIC/SVMClassification/
Google Search
University of New Orleans, Department of Computer Science
Home - The Biomedical Informatics Group at UNO
RIC Home - The Nanopore Biophysics/Cheminformatics Group at UNO
Channel Current Cheminformatics Architecture, Background, Literature
Blockade Spike Dedector
HMM based Kinetic Feature Extraction
AdaBoost Feature Selection
SVM Classification
Cluster-scan Kinetic Phase Recognition
Ab inito Gen Structure Identification
Discriminatory Resolution and Conformational properties of a set of DNA haripins
Antibody Conformational Change, Activation of Effector Function, and Antigen Binding
Transcription Factors, their binding sites, and analysis of the Transcriptome
Meet the group
Submit

SVM Classification

Feature Vector File: /svm-test/feature_vector_file

Training File: /svm-test/8GC9AT.train

Number of Fold for Cross Validation: 5

C: 5

Kernel: Gaussian

Sigma for Gaussian: 500

Tolerance: 0.01

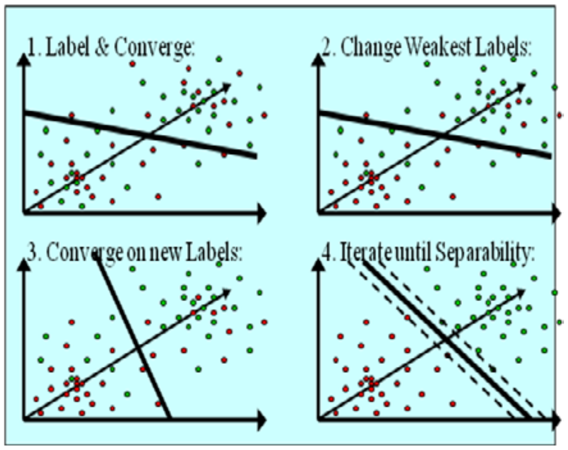
Epsilon: 0.01

Maximum Iteration: 10000

E-mail: Results will be sent to this address. meren@pardus.org.tr

3.9 External SVM-based Clustering

For standard channel current data, with standard HMM and SVM settings, initial external SVM-based clustering (Fig. 7 Left) of a synthetic mixture of two 9 base-pair DNA hairpins (such that labels are known for scoring) shows excellent clustering performance (see Fig. 7 Right).


1. Label & Converge:
2. Change Weakest Labels:
3. Converge on new Labels:
4. Iterate until Separability:

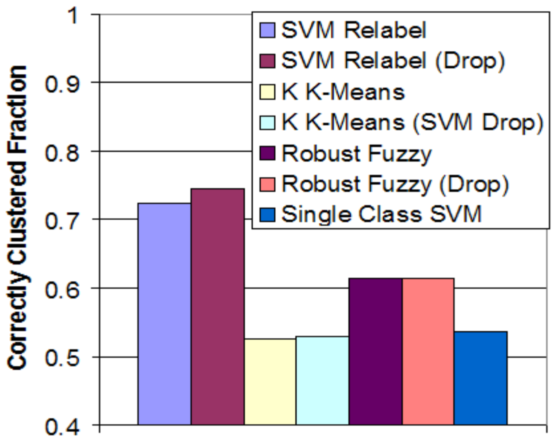


Fig. 7 Left. The External SVM clustering heuristic as shown, with intermediate convergence solutions, e.g., their hyperplane decision surfaces (a line) as shown. Right. Comparison of Clustering methods with external SVM clustering [25]. The strong performance of the external SVM clustering methods becomes apparent.

3.10 NTD ‘Binary’ event communication, a precursor to stochastic ‘phase’ modulation (SPM)

In the NTD experiments, the molecular dynamics of a (single) captured transducer molecule provide a unique stochastic reference signal with stable statistics on the observed, single-molecule blockaded, channel current, somewhat analogous to a carrier signal in standard electrical engineering signal analysis. Discernible changes in blockade statistics, coupled to SSA signal processing protocols, enable the means for a highly detailed characterization of the interactions of the transducer molecule with binding targets (cognates) in the surrounding (extra-channel) environment.

The transducer molecule is specifically engineered to generate distinct signals depending on its interaction with the target molecule. Statistical models are trained for each binding mode, bound and unbound, for example, by exposing the transducer molecule to zero or high (excess) concentrations of the target molecule. The transducer molecule is engineered so that these different binding states generate distinct signals with high resolution. Once the signals are characterized, the information can be used in a real-time setting to determine if trace amounts of the target are present in a sample through a serial, high-frequency sampling, and pattern recognition, process.

Thus, in NTD applications of the SSA Protocol, due to the molecular dynamics of the captured transducer molecule, a unique reference signal with stationary (or approximately stationary) statistics is engineered to be generated during transducer blockade, analogous to a carrier signal in standard electrical engineering signal analysis. The adaptive SSA machine learning algorithms for real-time analysis of the stochastic signal generated by the transducer molecule offer a “lock and key” level of signal discrimination. The heart of the signal processing algorithm is an adaptive Hidden Markov Model (AHMM) based feature extraction method, implemented on a distributed processing platform for real-time operation. For real-time processing, the AHMM is used for feature extraction on channel blockade current data, while classification and clustering analysis are implemented using a Support Vector Machine. In addition, the design of the machine learning based algorithms allow for scaling to large datasets, real-time distributed processing, and are adaptable to analysis on any channel-based dataset, including resolving signals for different nanopore substrates (e.g. solid state configurations) or for systems based on translocation technology. The machine learning software has also been integrated into the nanopore detector for “real-time” pattern-recognition informed (PRI) feedback [20,21] (see Fig. 10). The methods used to implement the PRI feedback include *distributed* HMM and SVM implementations, which enable the processing speedup that is needed.

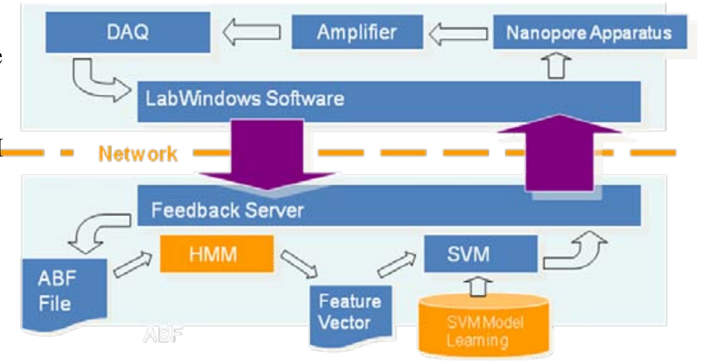


Figure 10. PRI Sampling Control (see [21] for specific details). Labwindows Feedback Server Architecture with Distributed CCC processing. The HMM learning (on-line) and SVM learning (off-line), denoted in orange, are network distributed for N-fold speed-up, where N is the number of computational threads in the cluster network.

A mixture of two DNA hairpin species (denoted {9TA, 9GC} in [9]) is examined in an experimental test of the PRI system [21]. In separate experiments, data is gathered for the 9TA and 9GC blockades in order to have known examples to train the SVM pattern recognition software. A nanopore experiment is then run with a 1:70 mix of 9GC:9TA, with the goal to eject 9TA signals as soon as they are identified, while keeping the 9GC's for a full 5 seconds (when possible, sometimes a channel-dissociation or melting event can occur in less than that time). The results showing the successful operation of the PRI system is shown in Fig. 11 as a 4D plot, where the radius of the event 'points' corresponds to the duration of the signal blockade (the 4th dimension). The result in Fig. 11 demonstrates an approximately 50-fold speedup on data acquisition of the desired minority species.

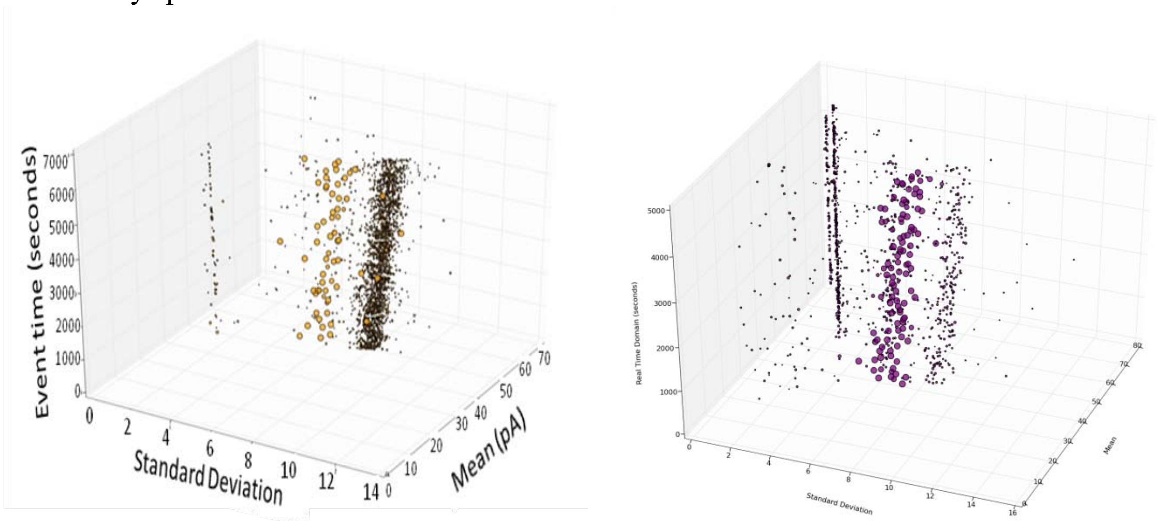


Figure 11. PRI Mixture Clustering Test with 4D plot [21]. Left. The vertical axis is the event observation time, and the plotted points correspond to the standard deviation and mean values for the event observed at the indicated event time. The radius of the points correspond to the duration of the corresponding signal blockade (the 4th dimension). Three blockade clusters appear as the three vertical trajectories. The abundant 9TA events appear as the thick band of small-diameter (short duration, ~100ms) blockade events. The 1:70 rarer 9GC events appear as the band of large-diameter (long duration, ~ 5s) blockade events. The third, very small, blockade class corresponds to blockades that partially thread and almost entirely blockade the channel. **Right. 1:280 rarer events.** Five-species pattern recognition informed sampling: greater than 99.9% accuracy inherited from off-line version

3.11 General SSA Protocol

The SSA Protocol tries to associate acquisition, feature extraction, classification, and clustering tasks with their most appropriate machine learning method, given the data, the noise properties, the operational time-constraints, and other constraints involved. Since data processing is often encountered in stages, the decomposition described in what follows is in terms of stages for acquisition, feature extraction, classification, and clustering, but the methods can have more complex seqneces of operation or embedded operation in another method, etc., as is described, to some extent, in what follows as well.

(Stage 1) primitive feature identification: this stage is typically finite-state automaton based, with feature identification comprising identification of signal regions (critically, their beginnings and ends), and, as-needed, identification of sharply localizable 'spike' behavior in any parameter of the 'complete' (non-lossy, reversibly transformable) classic EE signal representation domains: raw time-domain, Fourier transform domain, wavelet domain, etc. (The methodology for spike detection is shown applied to the time-domain in [7].) Primitive feature extraction can be operated in two modes: off-line, typically for batch learning and tuning on signal features and acquisition; and on-line, typically for the overall signal acquisition (with acquisition parameters set – e.g., no tuning), and, if needed, 'spike' feature acquisition(s). See [1] for details.

(Stage 2a) feature identification and feature selection: this stage in the signal processing protocol is typically Hidden Markov model based, where identified signal regions are examined using a fixed state HMM feature extractor or a template-HMM (states not fixed during template learning process where they learn to 'fit' to arrive at the best recognition on their train-data, the states then become fixed when the HMM-template is used on test data). The Stage 2 HMM methods are the central methodology/stage in the CCC protocol in that the other stages can be dropped or merged with the Stage 2 HMM in many incarnations. See [1] for details.

(Stage 3) classification: this stage is typically SVM based. SVMs are a robust classification method. If there are more classes to discern than two, the SVM can either be applied in a Decision Tree construction with binary-SVM classifiers at each node, or the SVM can internally represent the multiple classes, both are done in proof-of-concept experiments that are described. Depending on the noise attributes of the data, one or the other approach may be optimal (or even achievable). Both methods are typically explored in tuning, for example, where a variety of kernels and kernel parameters are also chosen, as well as tuning on internal KKT handling protocols. See [1] for details.

(Stage 4) clustering: this stage is often not performed in the 'real-time' operational signal processing task, as it is more for knowledge discovery, structure identification, etc., although there are notable exceptions, one such being the jack-knife transition detection, via clustering consistency with a causal boundary. This stage can involve any standard clustering method, in a number of applications, but the best performing in the channel current analysis setting is often found to be an SVM-based external clustering approach (see [25]), which is doubly convenient when the learning phase ends because the SVM-based clustering solution can then be fixed as the supervised learning set for a SVM-based classifier (that is then used at the operational level).

4 Results

4.1 Distributed HMM processing via 'Viterbi-overlap-chunking' with GPU speedup

Distributed processing has been done by use of simple chunking with overlaps 'sufficient' for recovery, where the details of the latter are described in what follows (and in [1]). A central Viterbi-like feature of the chinking methods employed is first described: the table chunking methods for the dynamic programming algorithms that have been developed involve only a single-pass computation analogous to the Viterbi algorithm (ignoring O(L) traceback). The Viterbi algorithm efficiently calculates the most probable state path. The Baum-Welch algorithm calculates the probability of having a state at a particular index, summing over all path probabilities that arrive at that state-instance, and is usually implemented as two passes, for the forward and backward parameters. In the Linear Memory HMM introduced in [26], however (see Methods),the Baum-Welch implementation has a distinctive trait other than a linear memory implementation, it's also a 'single-pass' implementation for the algorithm, which is needed for the Viterbi single-pass referenced, overlap-stitched, reconstituted signal in a distributed processing setting (see Methods for details). This can used be for brute force, and massively scalable, computational speed-up on all the HMM-based algorithms used in the SSA Protocol.

4.1.1 Single-Pass Table Algorithm

The table chunking methods for the dynamic programming algorithms that are described in what follows make use of a single-pass computation analogous to the Viterbi algorithm (ignoring O(L) traceback). In the Linear Memory HMM introduced in [26], and described below, the Baum-Welch algorithm implementation has a distinctive trait other than a linear memory implementation; itis also a 'single-pass' implementation.

4.1.2 Chunking with overlap resolution

In HMM signal processing latency becomes very prohibitive when attempting to increase device bandwidth or when input datasets are large. In [1], methods are described for performing HMM algorithms in a distributed manner by breaking into overlapping chunks and leveraging the Markovian assumption underlying the HMM to help arrive at a chunk-data reconstruction. The pathological instances where the distributed merges can fail to exactly reproduce the non-distributed HMM calculation can be made as least likely as desired with sufficiently strict, but not computationally expensive, segment join conditions. In this way, the distributed HMM provides a feature extraction that is equivalent to that of the sequentially run, general definition HMM, and with a speedup factor approximately equal to the number of independent CPUs operating on the data. The Viterbi most probable path calculation and the Expectation/Maximization (EM) calculation can both be performed in this distributed processing context.

The linear memory implementation described above (and in [27]) was optimized according to the observation that Viterbi traceback paths in the Viterbi procedure typically converge to the most likely state path and travel all together to the beginning of the decoding table – the picture being much like a river with minor tributaries backtracking onto that river, and maybe those 'tributaries' themselves have more minor state paths converging into them, etc. But the trait that is most notable in the convergence-durations to the 'main-tributary', or what is to be the most likely (Viterbi) path, is that it is usually a modest number of columns for many data types. This backwards Markovian memory loss on a tributary with respect to its origin (said to occur when backtracked and mixed with the main, Viterbi, convergence path of the tributaries) is hypothesized to be an indicator of the span of sequence needed to have Viterbi path probabilities in a given column that have settled into their properly ordered relative probabilities in that column. Further column processing refinement to bring the relative values of the Viterbi-path probabilities into better estimation is then possible. In distributed processing efforts, this “Viterbi relaxation time” is a key parameter that can be used to design an optimally overlapping chunking of the data sequence in a distributed speed-up on the sequence analysis.

A distributed signal processing test of some basic chunk reconstruction heuristics was performed on 5 computers with 300 signals. Each signal had 5000 samples. The resulting viterbi paths matched between the distributed HMM and standard HMM on a 10-column segment. For the standard HMM, EM training (5 loops) the Viterbi algorithm took 272 seconds. For distributed HMM with 5 CPU’s, the computational time was reduced to 69 seconds. So using 5 computers, we had a speedup of 3.94. A perfect de-segmentation was performed with an N=10 match window as indicated, initally, but it was found that a perfect re-stitching of segments was also possible simply with N=1 (see Fig. 12), due to the implicit stringency of the simultaneity condition (the overlap match, at the one position corresponding to N=1, must globally index to the same observation data index for both segments). The multi-chunk re-stitching makes use of the Viterbi path and the entire set of Viterbi traceback pointers in a given overlap set of columns [1].

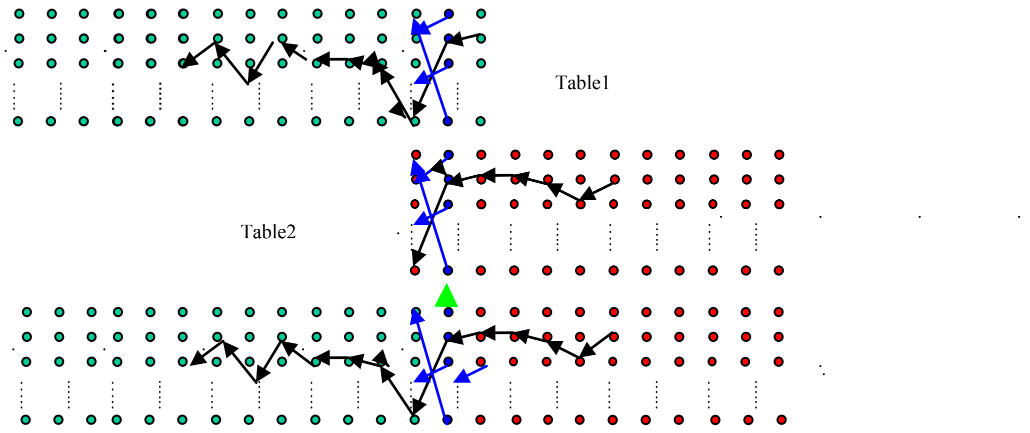


Figure 12. Viterbi column-pointer match de-segmentation rule. Table1 and Table2 are overlapped. And their blue columns have the same pointers. Then the index of this blue column becomes the joint. The black pointers form the final viterbi path.

5 Discussion

A brief discussion follows on the Ruby-n-Rails approach adopted in the web portal development (Sec. 5.1), followed by a synopsis of the NTD & SCW signal processing capabilities that are possible with the SSA Protocol (Sec. 5.2).

5.1 Web Portal Front-End via Ruby-on-Rails approach

Our web front-end for data collection and analysis is written in Ruby on Rails, an open source web application framework for the Ruby programming language. Ruby is a dynamic, reflective, general purpose OO programming language that supports many high level constructs such as blocks, closures, and dynamic reflection and alteration which allow for metaprogramming.

Ruby on Rails follows the Model-View-Control architecture pattern allowing for web applications to have a good separation of responsibilities between its related components. It also affords great freedom in rapid prototyping, allowing us to design a module and implement it quickly and efficiently.

The Ruby on Rails front-end web application running on a Mongrel web/application server with a MongoDB database backend. MongoDB is a high-performance, schema-free, document-oriented database that allows us to model data in a natural way. Since Ruby on Rails is a database-agnostic framework, migration to a different platform requires only changing a single database configuration file.

The application front-end provides a Graphical User Interface to a collection of data analysis tools including an acquisition FSA, a Nanopore Spike-Detector, SVM classification tools including Multiclass SVM, and numerous variations of the Hidden Markov Model (Prokaryotic HMM based gene predictor, and an HMM with Duration for gene structure identification). Rails provides an interface for all of these tools, written primarily in C and Perl, by allowing a user to upload the data that they want to be analyzed and choosing some parameter thresholds. The data and the parameters are then associated and stored in the database for analysis.

Once a user submits a dataset and chooses parameters, the application then utilizes the data collection tools mentioned above in the background, allowing the user to continue within the application uninterrupted. Once the job has completed, the user is notified via email of the results of our analysis. The results are also viewable through the GUI for comparison across multiple data runs or multiple parameter choices. We have geared this application toward the non-expert, by providing code to allow the user to normalize their data, as well as give helpful 'pre-set' information about the various parameters and tools to choose from.

By separating the GUI from the analysis tools, we have maximized efficiency in both application design, by using a framework more naturally suited for the web; and in throughput, by using highly efficient languages for the data analysis.

5.2 NTD signal analysis and SCW applications involving the SSA toolset

The NTD signal analysis, for example, demonstrates the simplest stochastic carrier wave utilization in a biophysics experimental setting – a stochastic phase modulation (with just two phases of stationary statistics). A minor elaboration on the signal analysis, to go from a simple two-state (bound/unbound) signal recognition to a two-phase SCW telegraph signal, then yields the rudimentary implementation for stochastic carrier communications purposes.

In SCW applications, the HMMD recognition of a transducer signal's stationary statistics has benefits analogous to 'time integration' heterodyning a radio signal with a periodic carrier in classic electrical engineering, where longer observation time is leveraged into higher signal resolution. In order to enhance such a 'time integration', or longer observation, benefit in the transducer signal, periodic (or stochastic) modulations may be introduced to the transducer environment. In a high noise background, for example, modulations may be introduced such that some of the transducer level lifetimes have heavy-tailed, or multimodal, distributions. With these modifications, a single transducer molecule signal could be recognizable in the presence of noise from many more channels than otherwise, providing an application for the SCW approach in nanopore transduction detection simply by leveraging existing signal processing capabilities with modern computers.

6 Conclusions

Web access portals are being developed for the Methods in the SSA Protocol. The Web Portals will provide access to Methods and Tools that can be used without an expert to perform sophisticated machine learning tasks and information processing architectures. If speed and real-time processing is needed, server access is being developed to access the Meta Logos distributed SSA implementations (and specialty tools).

In experimental efforts, stochastic sequential analysis methods provide a robust and efficient means to make a device or process as 'smart' as it can usefully be, with possible enhancement to device (or process) sensitivity and productivity and efficiency, as well as possibly enabling new capabilities for the device or process (via transduction coupling, for example, as with the nanopore transduction detector (NTD) platform). The SSA Protocol can work with existing device or process information flows, or can work with additional information induced via modulation or introduction via transduction couplings (comprising carrier references [22,23], among other things). Hardware 'device-awakening' and process-enabling may be possible via introduction of modulations or transduction couplings, when used in conjunction with the SSA Protocol implemented to operate on the appropriate timescales to enable real-time experimental or operational control.

In sequential data analysis and signal communications, hidden Markov models are a pervasive and fundamental tool. Critical HMM tools have recently been improved via a number of computationally efficient generalizations. This has led to the SSA Protocol and associated methods, as described here, for a new kind of signal processing capability. In particular, a stochastic carrier-wave (SCW) method is now possible. All of these methods are currently inaccessible to the non-expert even thought their application is straightforward and their result is generally very impressive. To address this need, web interfaces and streaming-data server interfaces are proposed and initial work along these lines has been successful, as described in the Methods. To meet the needs of large dataset training, and streaming data (on-line) processing, distributed SSA Protocol methods have been developed, and successful results along these lines are outlined in the Results.

7 Acknowledgements

The authors would like to thank lab technicians Amanda Alba and Eric Morales for help performing the nanopore experiments and University of New Orleans student Evenie Horton for help with the nanopore experiments and the channel current cheminformatics analysis. SWH would like to thank the University of New Orleans, NIH, NSF, NASA, and the Louisiana Board of Regents for research support. SWH would also like to thank

META LOGOS Inc., for research support and a research license. (META LOGOS was co-founded by SWH in 2009, when it obtained exclusive license to the NTD and machine-learning based signal processing intellectual property.)

8 References

[1] Winters-Hilt, S. and R. Adelman. Method and System for Characterizing or Identifying Molecules and Molecular Mixtures. USPTO Filing. Meta Logos Inc. 2010.

[2] Winters-Hilt, S. and C. Baribault. A Meta-state HMM with application to gene structure identification in eukaryotes. EURASIP Journal of Advances in Signal Processing, Special Issue, Genomic Signal Processing, 2010.

[3] Winters-Hilt, S., Jiang, Z., and C. Baribault. Hidden Markov model with duration side-information for novel HMMD derivation, with application to eukaryotic gene finding. EURASIP Journal of Advances in Signal Processing, Special Issue on Genomic Signal Processing, 2010.

[4] Winters-Hilt S and Jiang Z. A hidden Markov model with binned duration algorithm. IEEE Trans. on Sig. Proc., Vol. 58 (2), Feb. 2010.

[5] Roux B and Winters-Hilt S. Hybrid SVM/MM Structural Sensors for Stochastic Sequential Data. BMC Bioinf. 9 S9, S12 (2008).

[6] Iqbal R, Landry M, Winters-Hilt S: DNA Molecule Classification Using Feature Primitives. BMC Bioinformatics 2006, 7 S2: S15.

[7] Landry M, Winters-Hilt S. Analysis of nanopore detector measurements using machine learning methods, with application to single-molecule kinetic analysis. BMC Bioinf. 8 S7, S12 (2007).

[8] Donoho D. Compressed sensing. IEEE Trans. On Information Theory, 52(4), pp. 1289 - 1306, April 2006.

[9] Winters-Hilt, S., W. Vercoutere, V. S. DeGuzman, D. Deamer, M. Akeson, and D. Haussler, "Highly Accurate Classification of Watson-Crick Base-Pairs on Termini of Single DNA Molecules," *Biophys. J.* Vol. 84, pg 967, 2003.

[10] Winters-Hilt S: Hidden Markov Model Variants and their Application. BMC Bioinf. 2006, 7 S2: S14.

[11] Vapnik, V. N. 1999. The Nature of Statistical Learning Theory (2nd Ed.). Springer-Verlag, New York.

[12] J.D. Ferguson. Variable duration models for speech. Proceedings of Symposium on the Application of Hidden Markov models to Text and Speech, pages 143-179, 1980.

[13] L.R. Rabiner. A tutorial on hidden markov models and selected application in speech recognition. Proceedings of the IEEE, 77:257-286, 1989.

[14] P. Ramesh and J.G. Wilpon. Modeling state durations in hidden markov models for automatic speech recognition. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 1:381-384, 1992.

[15] SZ Yu and H. Kobayashi. An efficient forward-backward algorithm for an explicit-duration hidden markov model. IEEE Signal Processing Letters, 10:11-14, 2003.

[16] M.T. Johnson. Capacity and complexity of hmm duration modeling techniques. IEEE Signal Processing Letters, 12:407-410, 2005.

[17] Z. Ghahramani and M. Jordan. Factorial hidden markov models. Machine Learning, 29:245-273, 1997.

[18] Y. Singer S. Fine and N. Tishby. The hierarchical hidden markov model: Analysis and applications. Machine Learning, 32:41, 1998.

[19] K. Murphy and M. Paskin. Linear time inference in hierarchical hmms. Proceedings of Neural Information Processing Systems, 2001.

[20] Baribault, C and Winters-Hilt S. A novel, fast, HMM-with-Duration implementation -- for application with a new, pattern recognition informed, nanopore detector. BMC Bioinformatics 2007, 8 S7: S19.

[21] Eren AM, Amin I, Alba A, Morales E, Stoyanov A, and Winters-Hilt S. Pattern Recognition Informed Feedback for Nanopore Detector Cheminformatics. Accepted paper in book “Advances in Computational Biology”, Springer: Advances in Experimental Medicine and Biology, June 2010

[22] Winters-Hilt, S., and Pincus, S. Nanopore-based biosensing. PATENT, UNO filing, 2004.

[23] Winters-Hilt, S. , and Pincus, S. Channel current cheminformatics and bioengineering methods for immunological screening, single-molecule analysis, and single molecular-interaction analysis. PATENT, UNO filing, 2005. [All paragraph citations specifically refer to version PCT WO2006041983A2 in case of ambiguity.]

[24] Winters-Hilt, S.; U.S. Provisional Patent Application No. 61/233,732. Title: A Hidden Markov Model With Binned Duration Algorithm (HMMBD). August 13, 2009. Re-filing: Winters-Hilt, S.; U.S. Provisional Patent Application No. 61/234,885. Title: An Efficient Self-Tuning Explicit and Adaptive HMM with Duration Algorithm. August 18, 2009.

[25] Winters-Hilt S, Yelundur A, McChesney C, Landry M: Support Vector Machine Implementations for Classification & Clustering. BMC Bioinformatics 2006, 7 S2: S4.

[26] Churbanov, Alexander and S. Winters-Hilt. Implementing EM and Viterbi algorithms for Hidden Markov Model in linear memory. BMC Bioinformatics 2008, 9:228.

[27] Miklos I, Meyer I: A linear memory algorithm for Baum-Welch training. *BMC Bioinformatics* 2005, 6(231).