Distributed Machine Learning for Bioinformatics and Cheminformatics

Stephen Winters-Hilt

Bioinformatics Group at Computer Science Department, UNO:

- **Carl Baribault, Zuliang Jiang, Sam Merat, Murat Eren, Ken Armond, Alex Lu, Hang Zhang, Brian Roux, and Dr. Alexander Churbanov**

Nanopore Biophysics/Cheminformatics Group at Children's Hospital:

- **Eric Morales, Iftekhar Amin, Amanda Alba, Karen Thomson, Molly Oehmichem, and Dr. Alexander Stoyanov**

Distributed Data Sharing

<u>Data Warehousing: (1-10 TB long-term)</u>

Channel Current (and other power signal) Data Repository (~ 1TB)

Genome Repository (prokaryotic and eukaryotic) (~ 10TB)

<u>Data Mining: (additional 1-10 TB short-term)</u>

High Order sub-sequence interpolated Markov Model Construction

Distributed Hidden Markov Model Processing

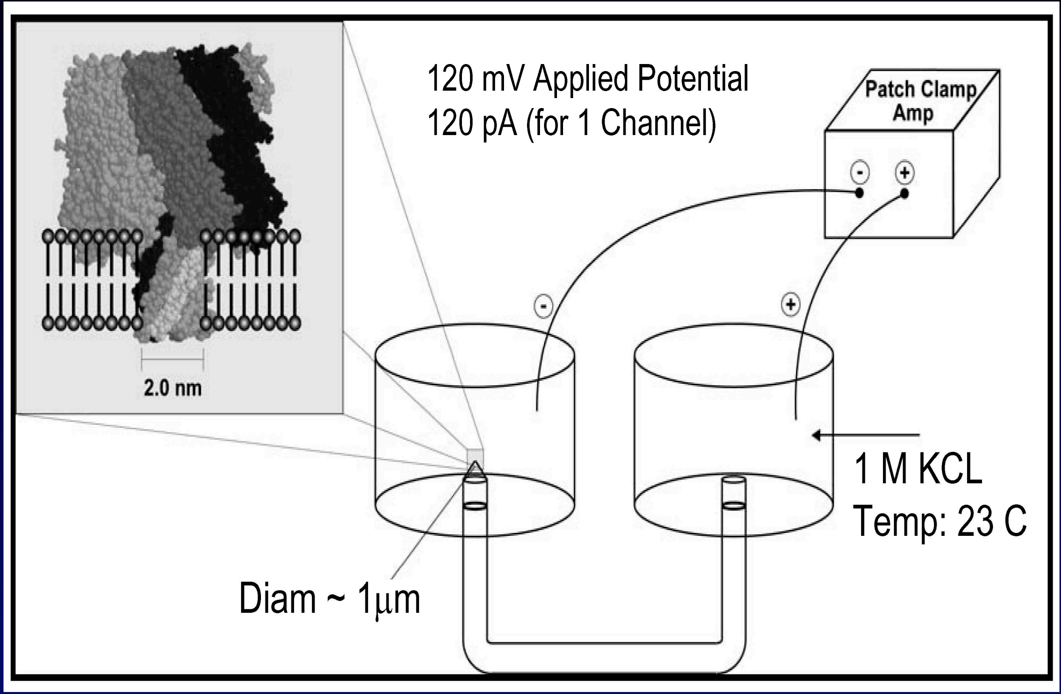
Distributed SVM Chunk Processing

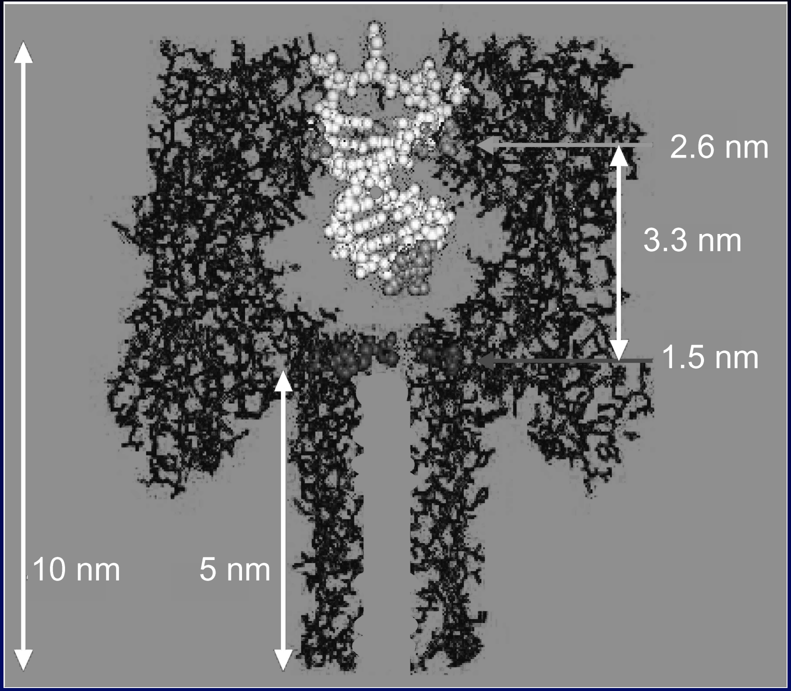
<u>Application Areas:</u>

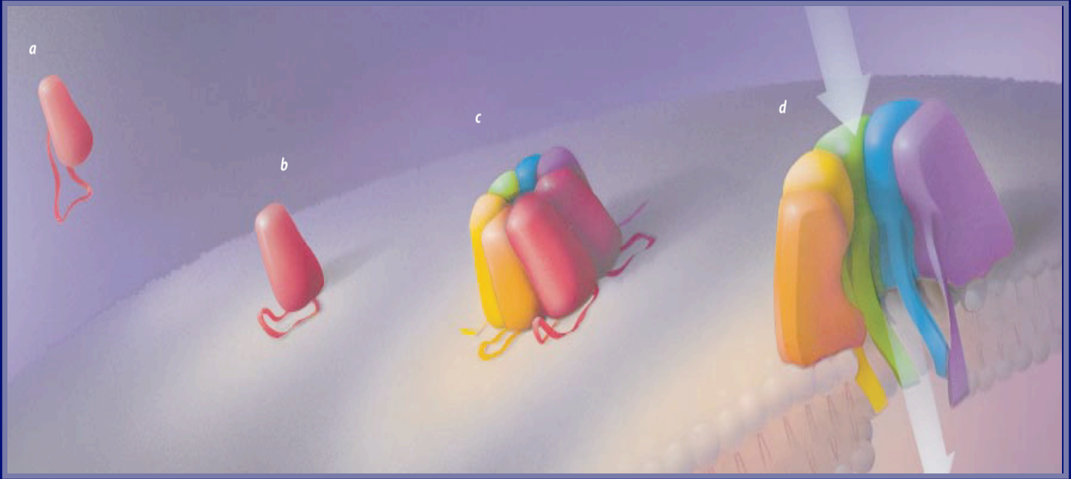
Cheminformatics -- enables Nanopore Detector capabilities

Bioinformatics -- used for gene-structure identification (including regulatory regions) and comparative genomics

The α-Hemolysin Nanopore Detector







Hagan Bailey, Sci. Am.

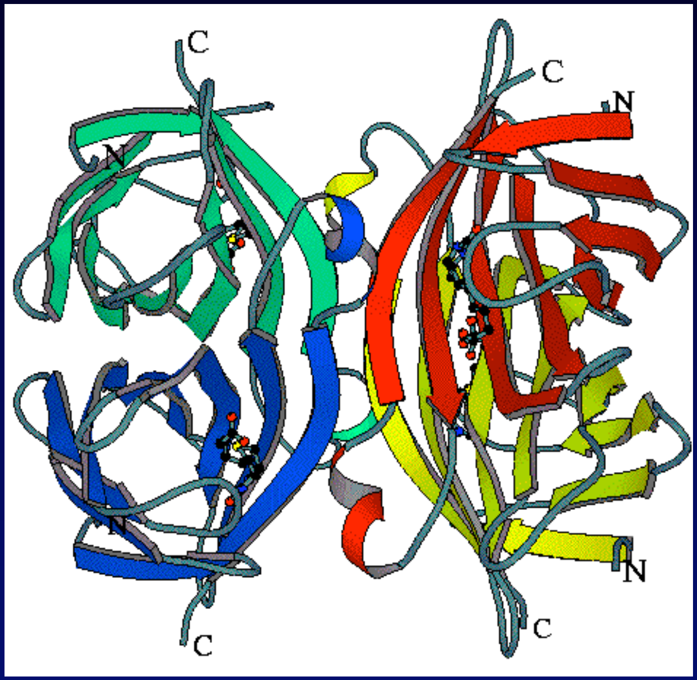
α-Hemolysin with a 9bp DNA hairpin

Nanopore Conception:
J.J. Kasianowicz; S. Bezrukov, A. Parsegian; I. Vodyanoy; D. Branton; D. Deamer; M. Akeson; H. Bailey; …

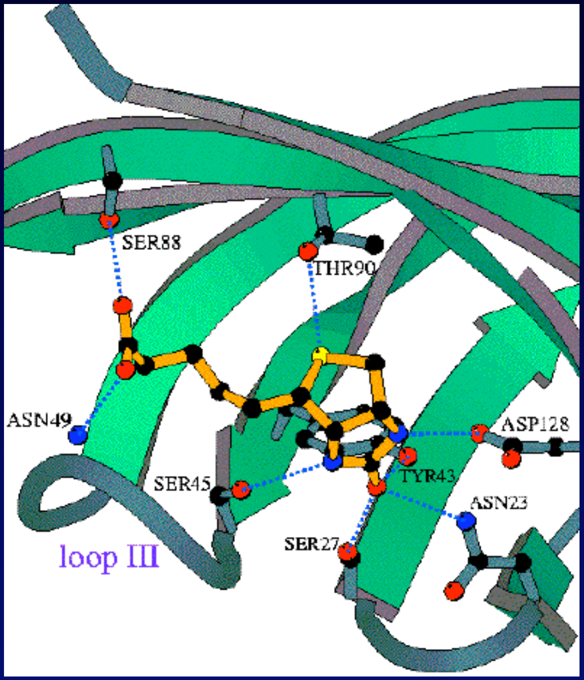
α-Hemolysin self-assembles from solution soluble monomers

The Streptavidin and Biotin Interaction: Ka~$10^{14}M^{-1}$

Tetrameric Streptavidin:



Hydrogen Bonding with Biotin:

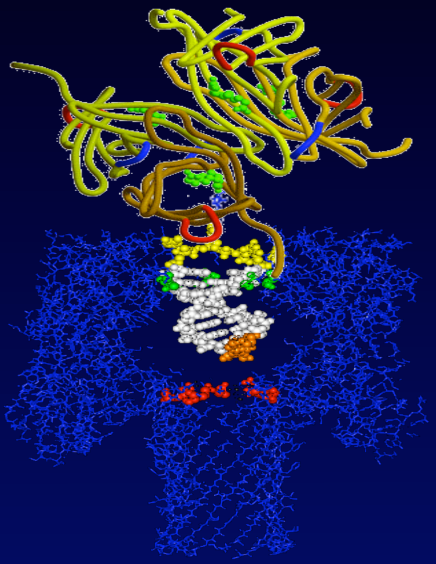


Streptavidin: 53,000 Daltons
Near Neutral pI

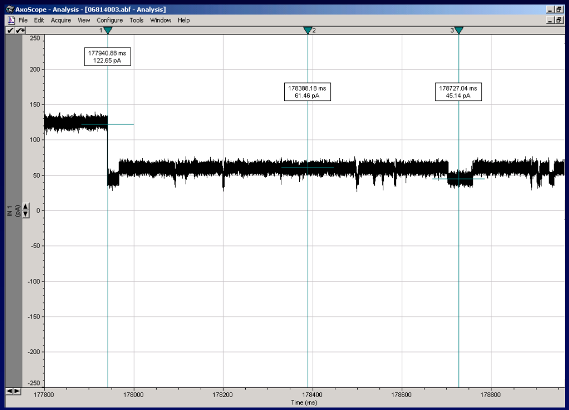
Biotin: 244.31 Daltons

S. Freitag, I. Le Trong, L. Klumb, P.S. Stayton, R.E. Stenkamp; Structural Studies of the Streptavidin Binding Loop; *Protein Science* **6** (1997), 1157 - 1166.

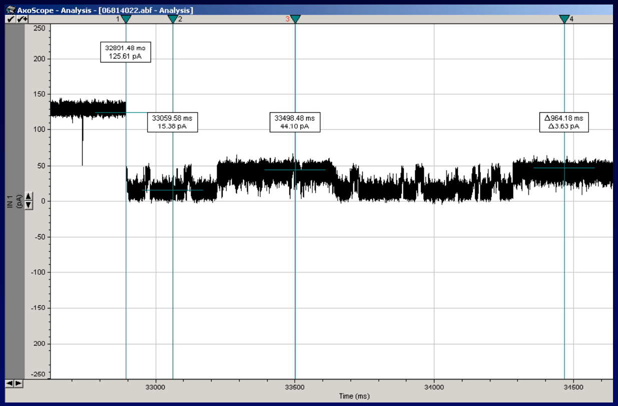
Tetrameric Streptavidin binding to a Biotinylated DNA Hairpin (9gc-Biot) Captured in the Nanopore Detector:



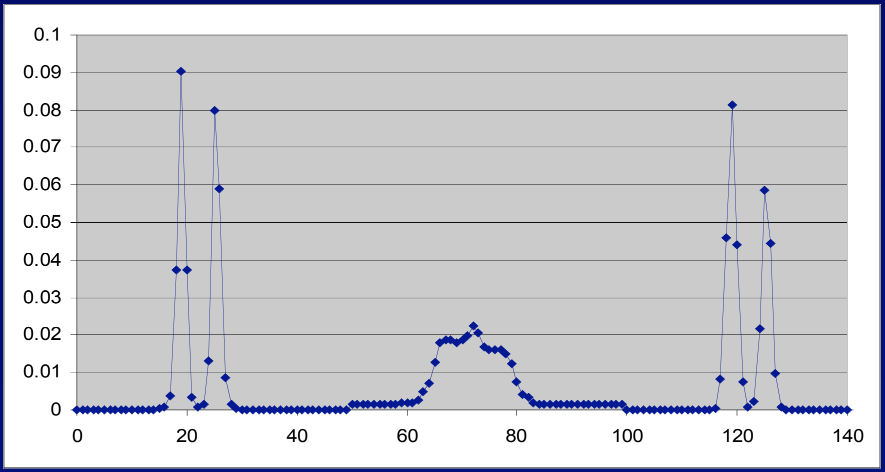
Biotinylated DNA Hairpin



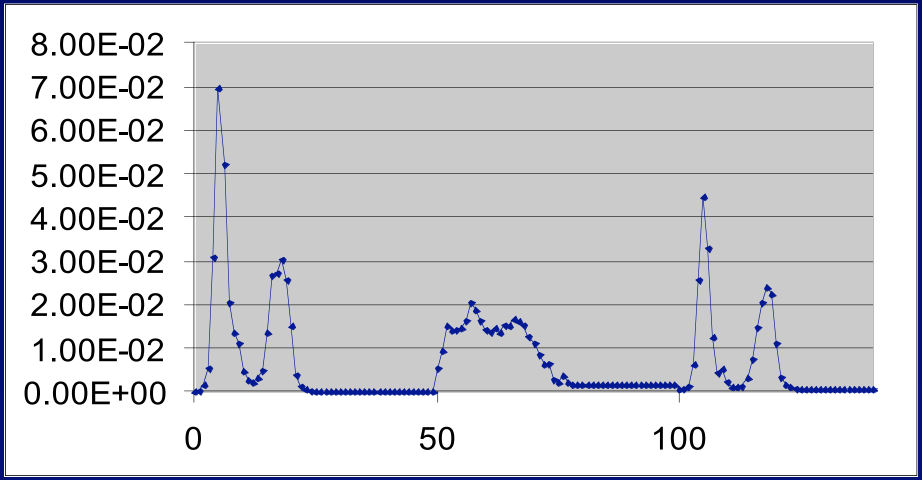
Streptavidin Binding



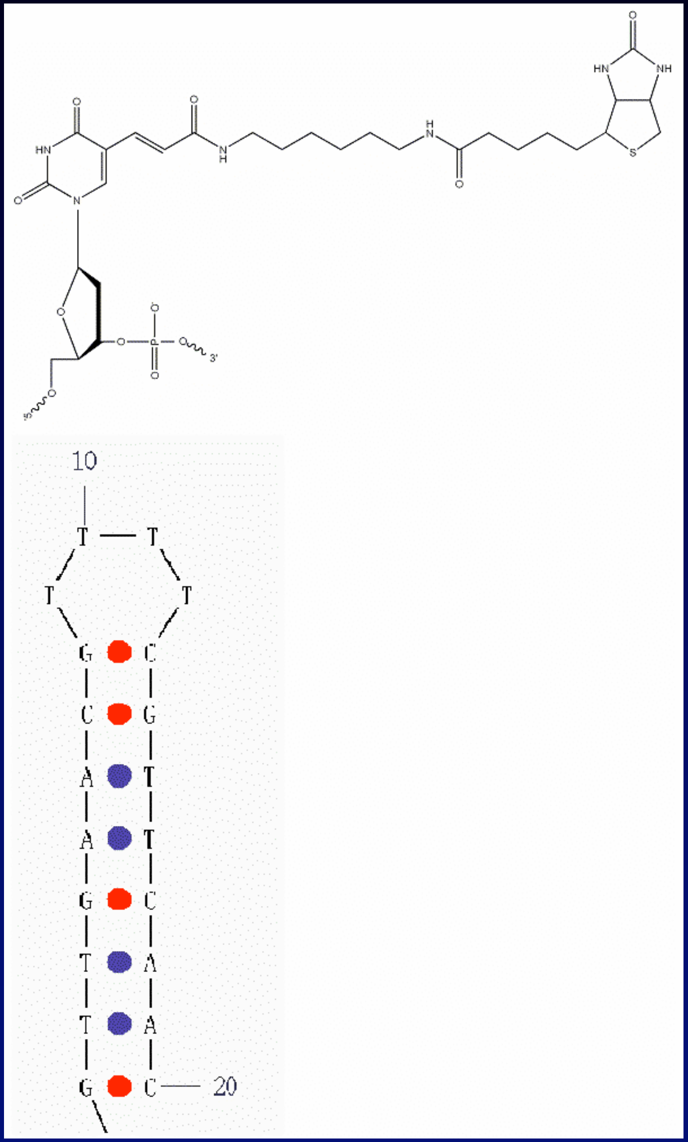
Biotinylated DNA Hairpin



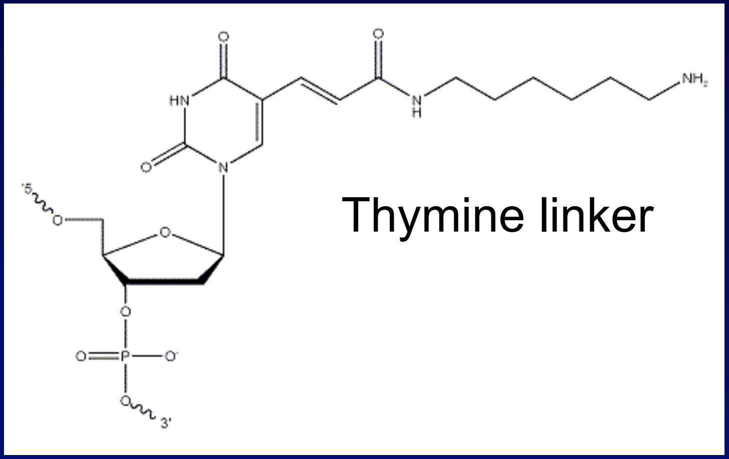
Streptavidin Binding

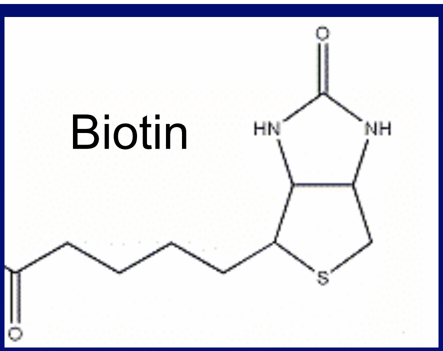


Modified DNA Hairpin

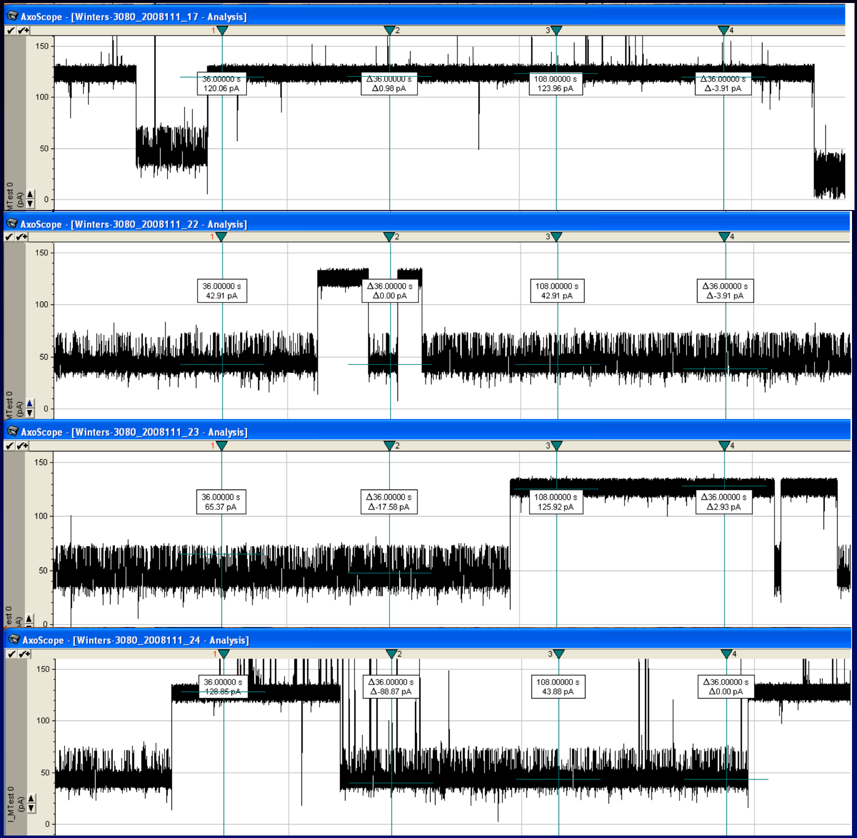


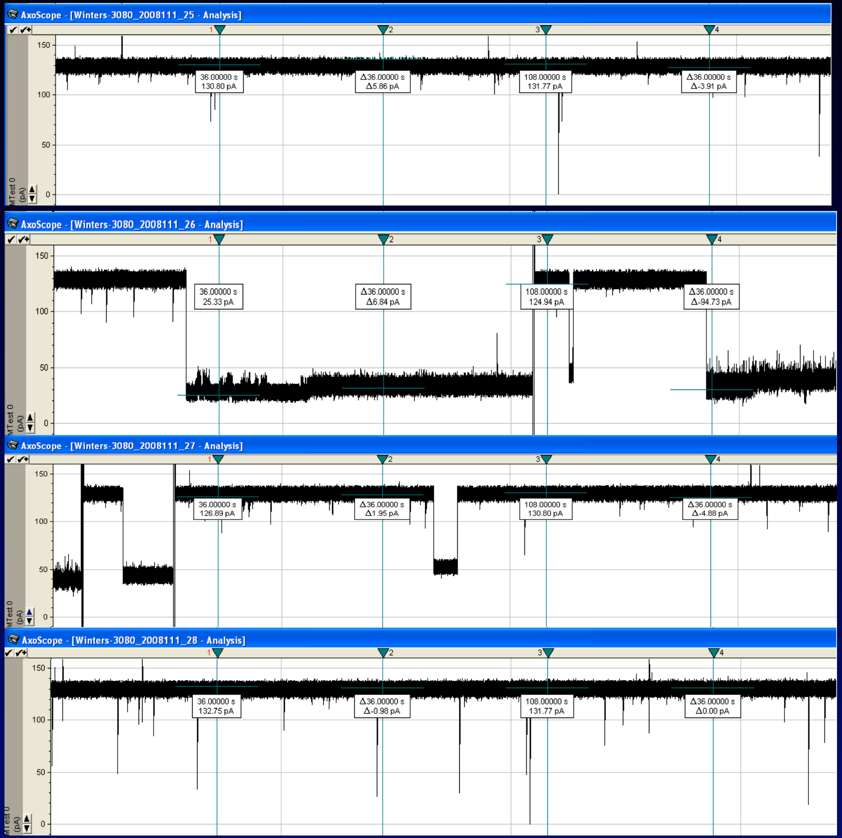
Our DNA hairpin has eight base pair stem region terminating in GC and a four Thymine loop. An internal DNA modification is added at the 10th base using a modified Thymine with a six carbon linker.





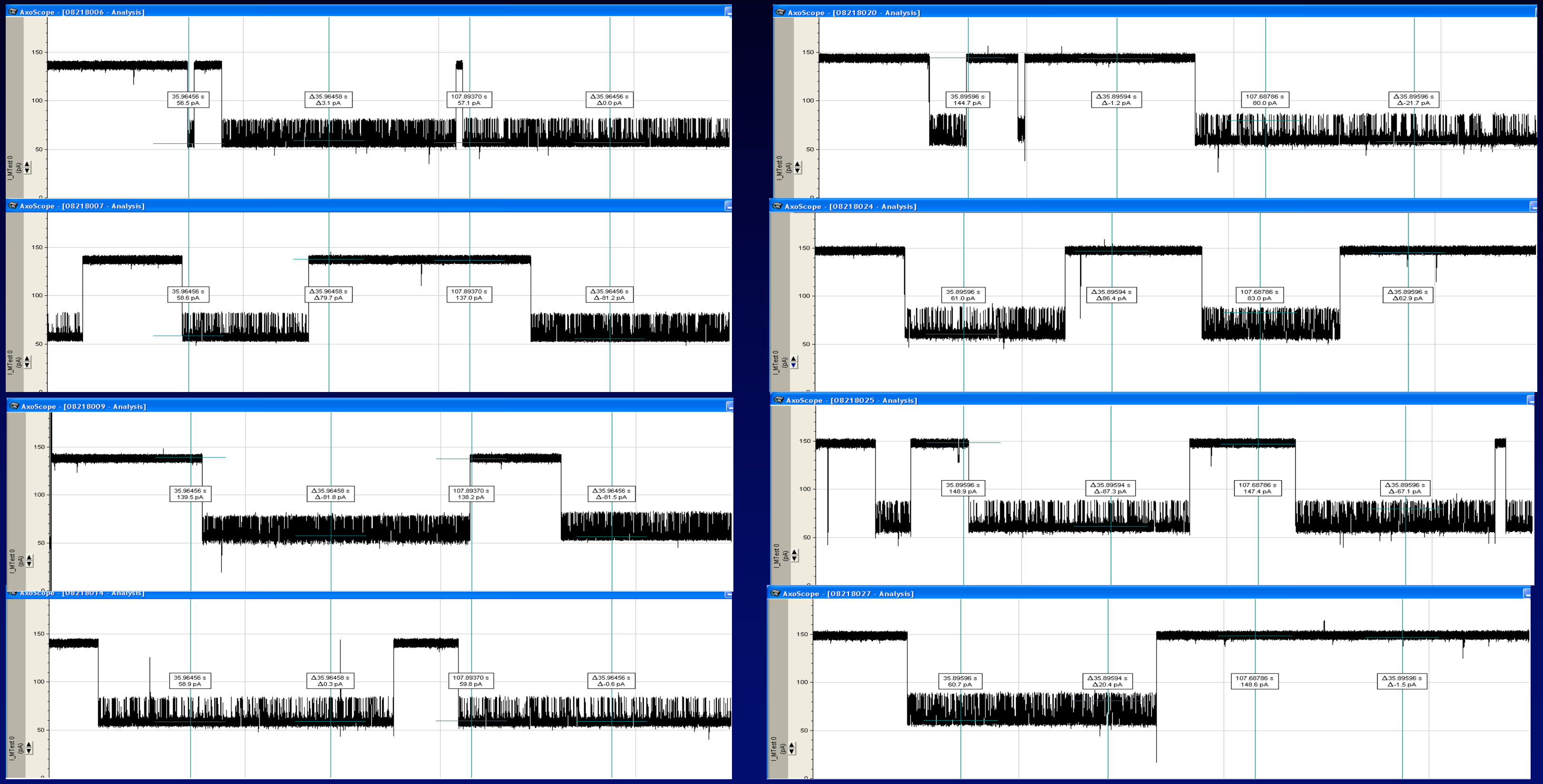
Bt-DNAhp Signal Suppression with Streptavidin Binding



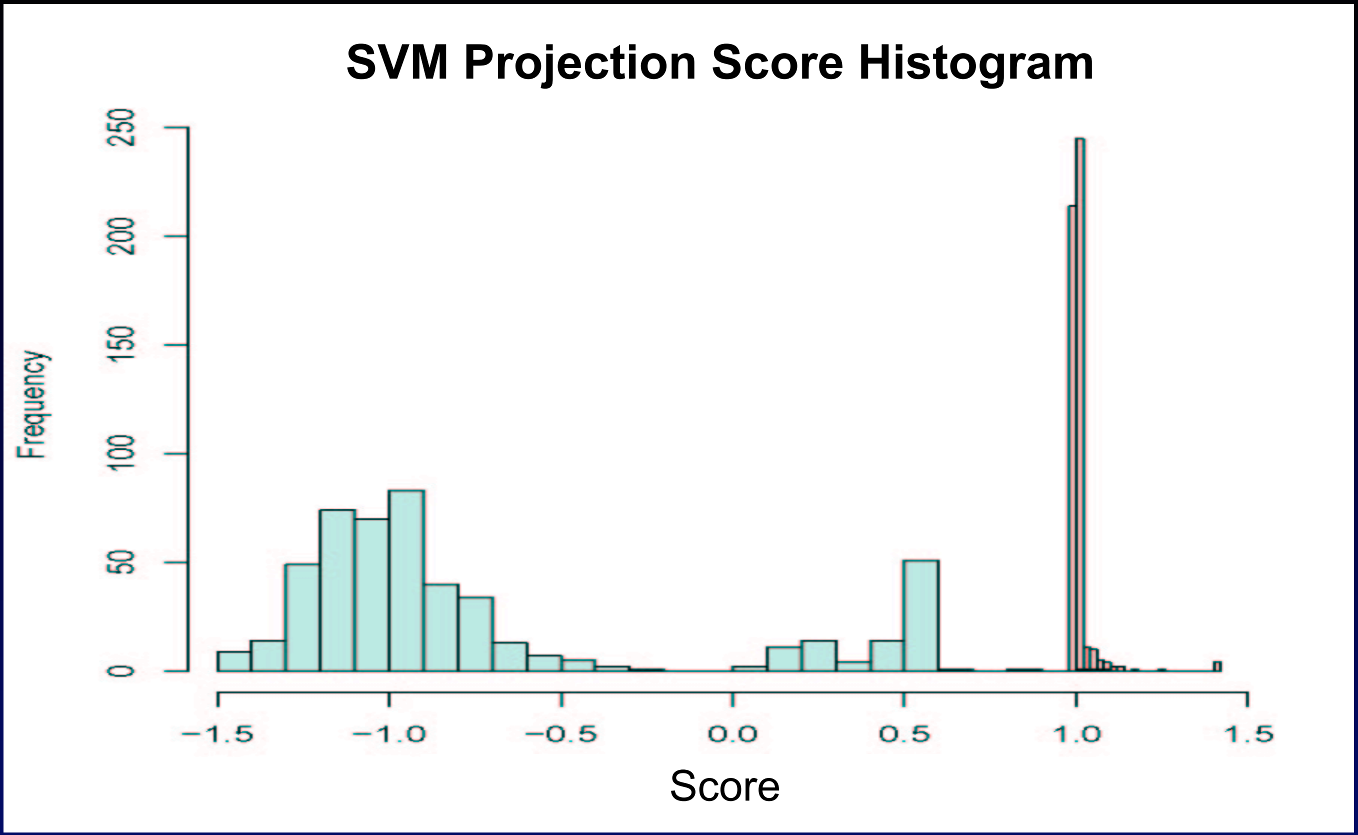


The left column is a series of channel's current reading (each three minute in length) displaying capture events of BiodT-DNA HP. On the right we show a decrease of these capture events as a result of an increase in Streptavidin concentration. Note the change in signal itself. Molar ratio of Streptavidin to hairpin to 2:1.

Negative-Specificity Control: Mixtures of "-DNAhp" molecules and Streptavidin

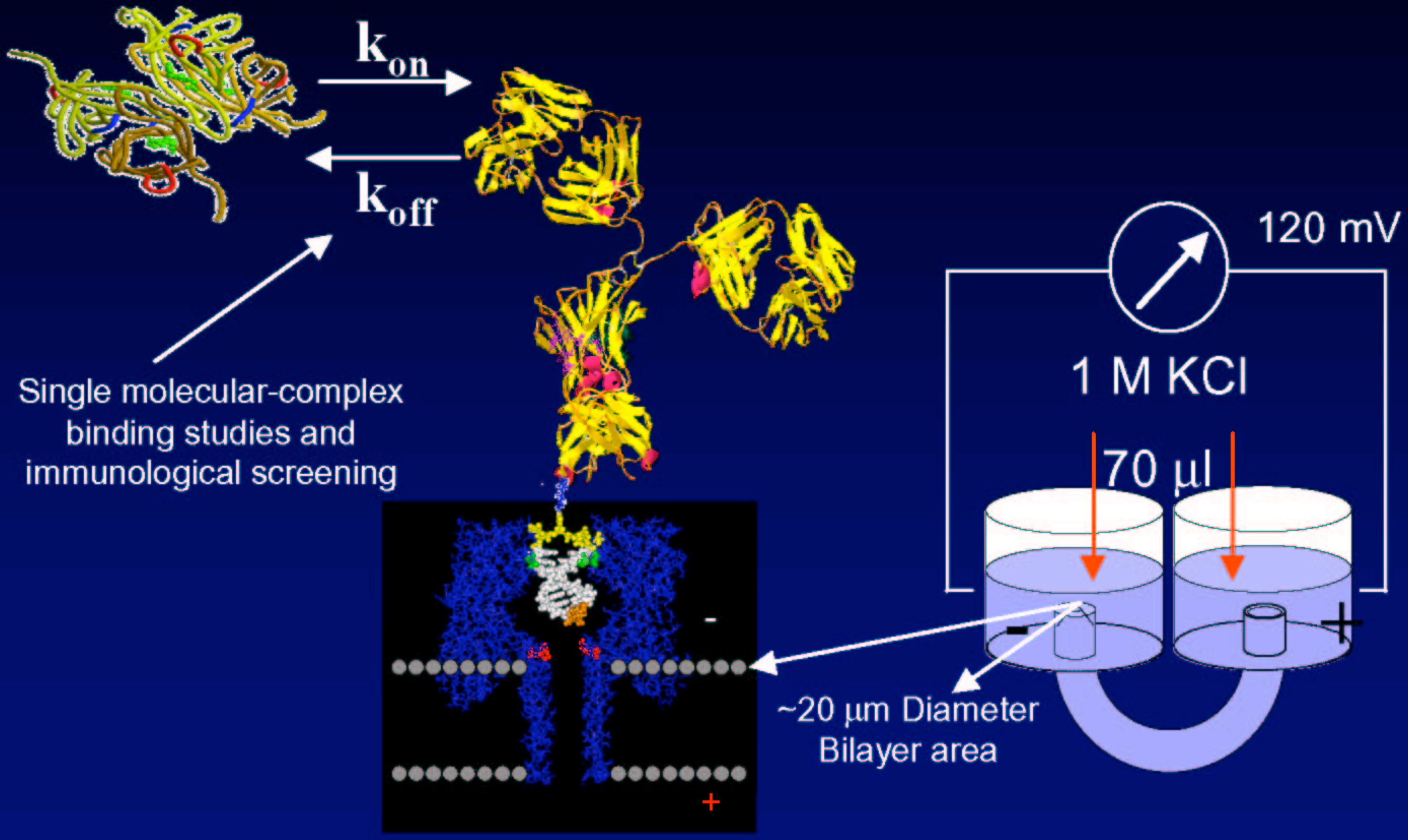


The left column is a series of observations from a negative-specificity control experiment involving "-DNHhp" before addition of Strepatvidin, the right column after addition of streptavidin. Each signal trace is the signal observed during a 3-minute interval. "-DNHhp" has the same structure of our biotinylated 8GC hairpin only without the biotin (just the six carbon linker). Upon addition of Streptavidin no difference in capture rates or change in toggle signal were observed (at equal streptavidin concentrations as used in previous experiments).



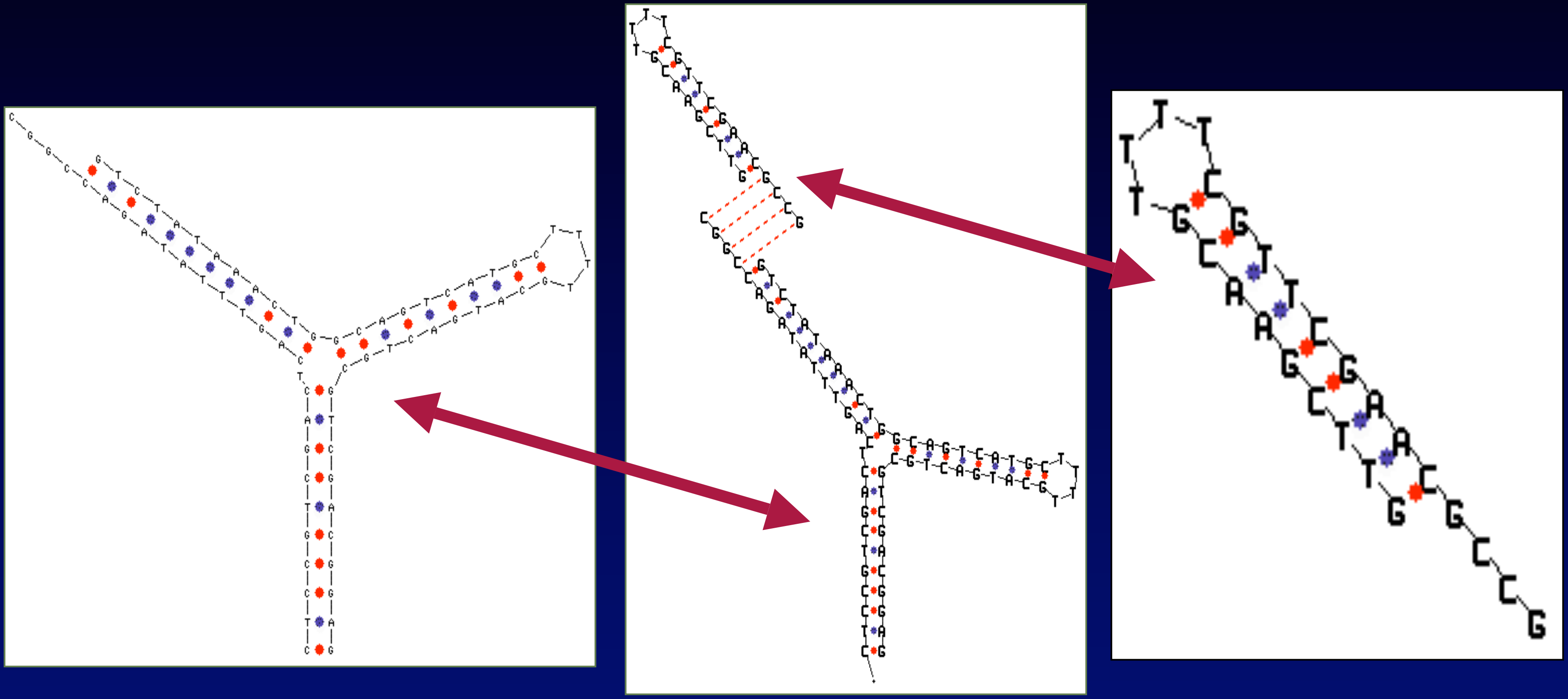
Cluster identification and counting via a SVM projection-score histogram. (This corresponds to SVM-External Clustering in Decision Space). Biotinylated hairpin signals comprise the positives. appearing as the large peak scoring around 1.0. The mixture signals seen after introduction of streptavidin are shown as the light blue bars. The score-clustering at 0.5 in the projection-score histogram corresponds to (unbound) biotinylated DNA hairpin signals that are successfully projected towards their corresponding signals in the positives. The other, clear, negative signals (in light blue), that score around −1.0, are hypothesized to correspond to the streptavidin-bound biotinylated DNA hairpins.

The α-Hemolysin transduction detector



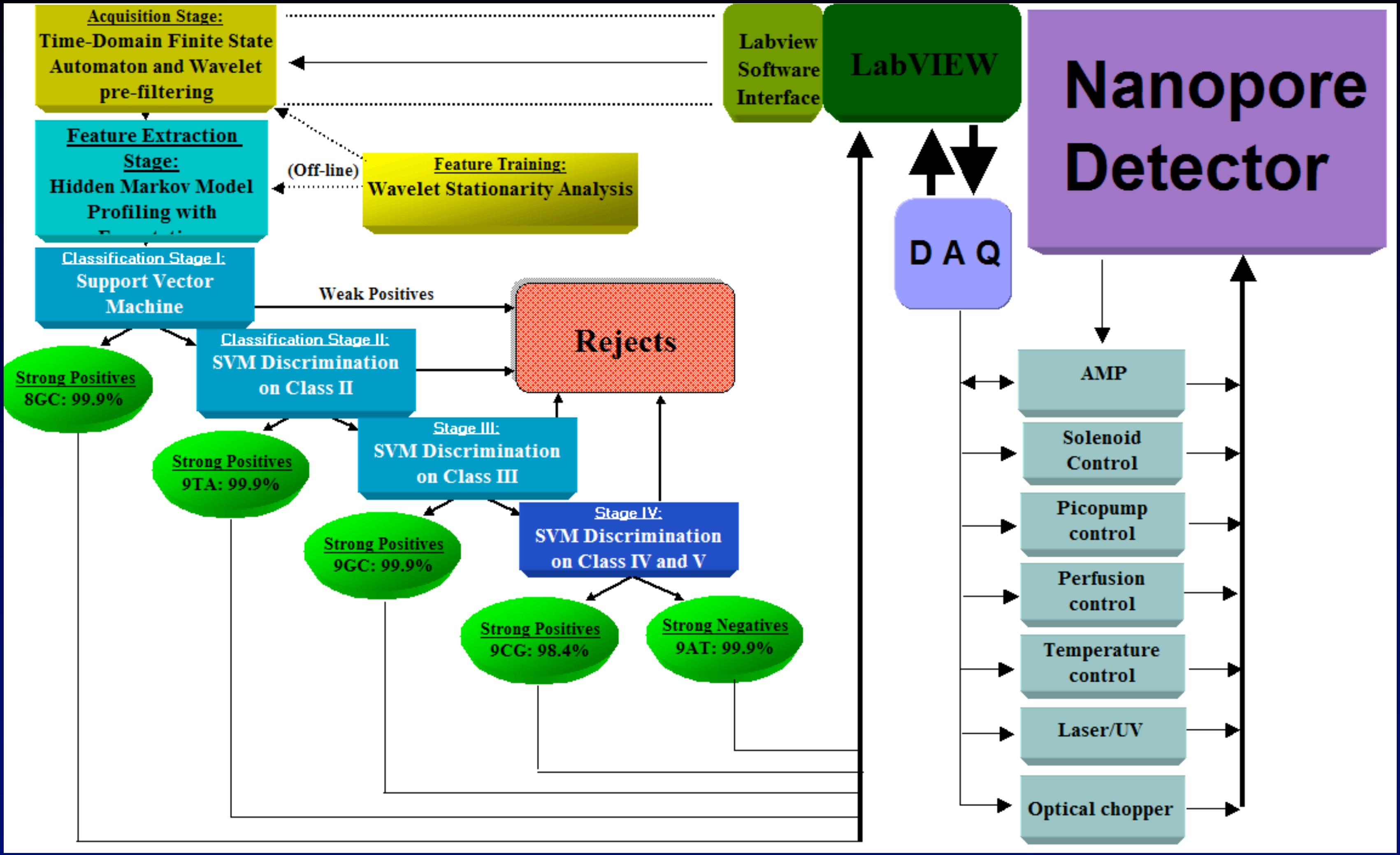
k_{on}
k_{off}
Single molecular-complex binding studies and immunological screening
120 mV
1 M KCl
70 µl
-
+
~20 µm Diameter Bilayer area

GC4-DNA Y-shaped Aptamer



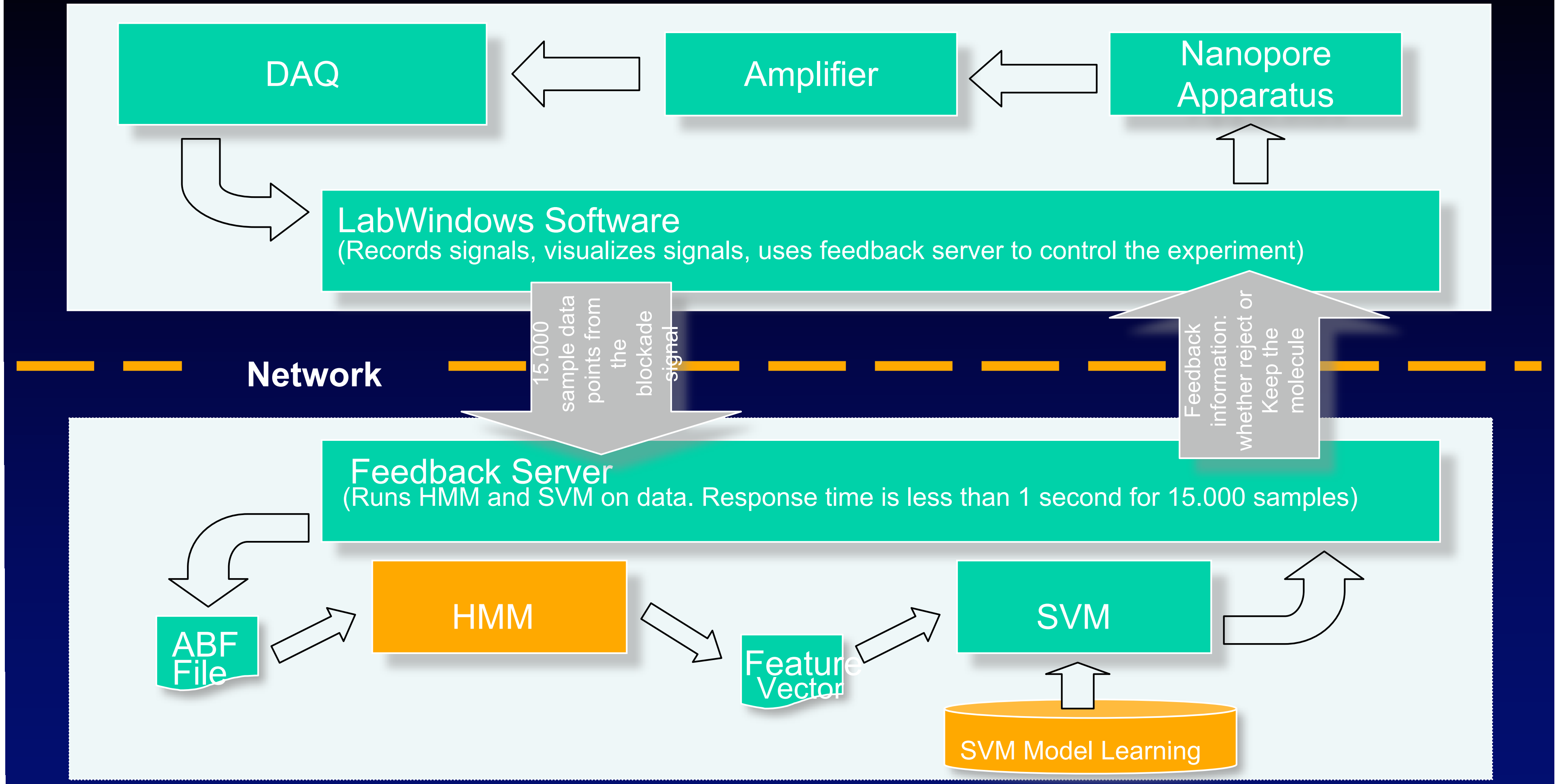
Our bifunctional Y-shaped aptamer with 5’-CGGC-3’ overhang (left), hairpin with complementary overhang (right) and in the middle complement base pair annealing to form composite molecule (center). The base of the Y shape in the event-transduction terminus that inserts into the alpha hemolysin channel to produce the blockade signal.

Nanopore Cheminformatics & Control Architecture



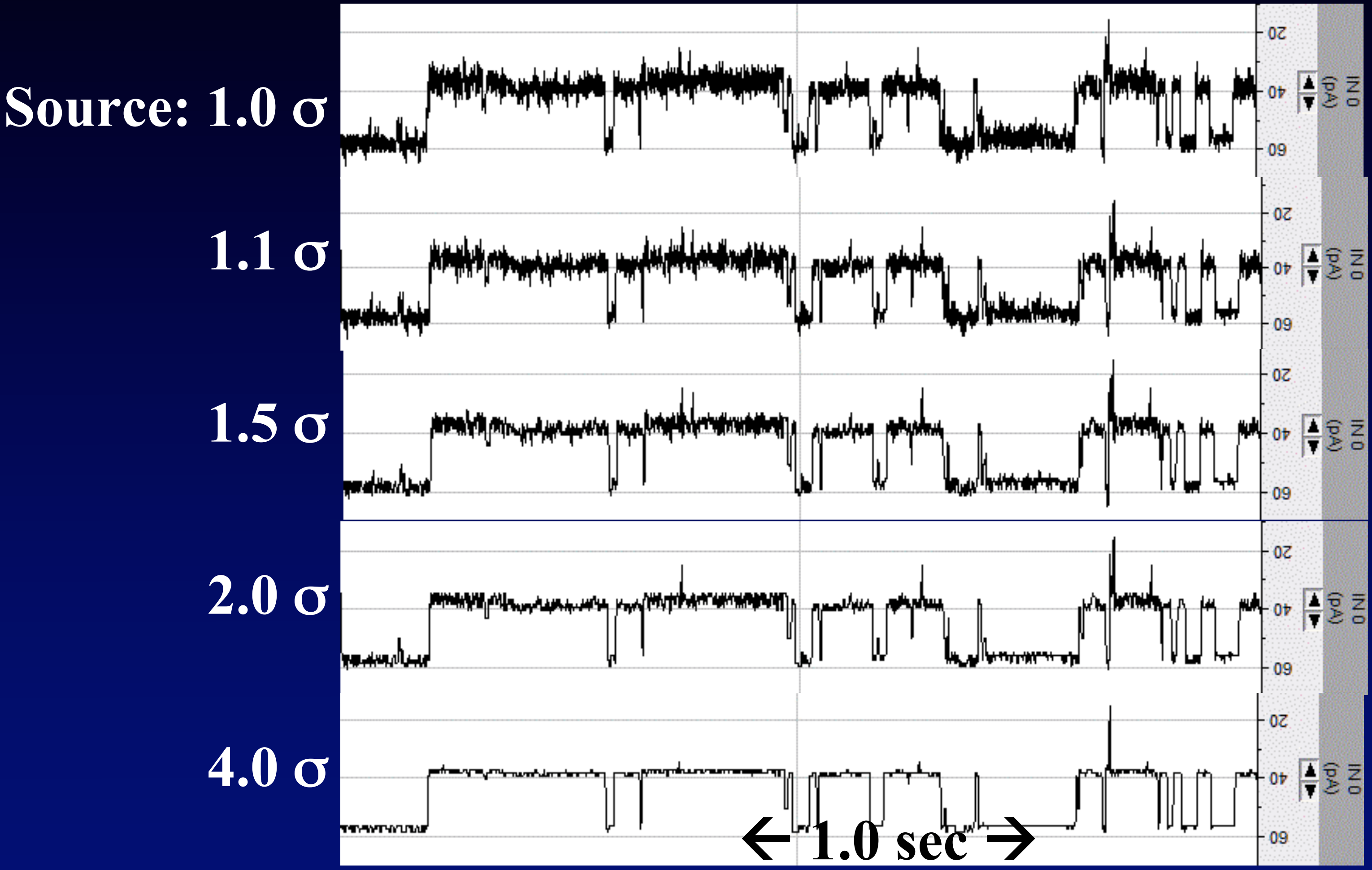
LabWindows Server now used. Data sent to cluster of Linux Clients via TCP/IP channel. Linux clients run expensive HMM analysis as distributed processes (similarly for off-line SVM training). The sample classification is used by the Server to provide feedback to the nanopore apparatus to increase the effective sampling time on the molecules of interest (this can boost nanopore detector productivity by magnitudes).

Real-time Channel Current Cheminformatics


DAQ
Amplifier
Nanopore Apparatus
LabWindows Software
(Records signals, visualizes signals, uses feedback server to control the experiment)
15.000 sample data points from the blockade signal
Feedback information: whether reject or Keep the molecule
Network
Feedback Server
(Runs HMM and SVM on data. Response time is less than 1 second for 15.000 samples)
ABF File
HMM
Feature Vector
SVM
SVM Model Learning

Labwindows/Feedback Server Architecture with Distributed CCC processing. A capture signal generated with the nanopore apparatus is filtered and amplified before it is sent through the DAQ. The Data AcQuisition device converts the analog signal to digital format for use in the display and recording of data in binary Axon (Molecular Devices) format. In the pattern recognition feedback loop, the first 200 ms detected after drop from baseline are sent via TCP-IP protocol to the HMM software, which generates a profile for each signal sent. The HMM-generated profile is processed with the SVM classifier to compare the real-time signal with previous training data in order to determine whether the signal is acceptable. The HMM learning (on-line) and SVM learning (off-line), denoted in orange, are network distributed processes for N-fold speed-up, where N is the number of computational threads in your cluster network.

HMM/EM EVA (Emission Variance Amplification) Projection for simplified tFSA Kinetic Feature Extraction:



Kinetic feature enhancement via a novel HMM/EM filter that "projects" via a gaussian parameterization on emissions with variance boosted by the factor indicated.

HMM with Duration

The HMM-with-duration (HMMwD) is an HMM that directly models the “true” sub-blockade duration probabilities, and provides a strong link to the underlying kinetic (physical) information that is desired (an EM optimization can be directly performed to yield the best estimate of the probability distributions on state durations. The means of those distributions, the kinetic half-lives, directly relate to the underlying kinetic coefficients). HMMwD is parameterized by the internal HMM signal representation (the emission and transition probabilities, and the duration distributions on state lifetimes), and can be efficiently implemented. With HMM-with-duration, feature extraction is more robust on long-lifetime states.

Novel, exact, HMMwD for EM and Viterbi

Standard HMM:

$p(d = x) = (a_{ii}^{\,x-1})\,(1-a_{ii})$.

Restricted to the geometric distribution.

New HMMwD:

$p(d = x) = (\prod_{i=1..x-1} p(d \geq i+1)/p(d \geq i))\,(1-p(d \geq x+1)/p(d \geq x))$.

This formula’s advantage is the calculation of p(d) can be distributed among the x consecutive steps, and it provides the exact distribution.

New HMM Table construction uses carry-sum cells for each state, with the new HMMwD p(d = x) definition. Computational time increases by a factor of D/N +1, where N=number of HMM states, D=number of bins in the length distribution representation, 1000 is used. If the number of states > 1000, then the factor is approx. 1! This provides a 1,000,000 speedup factor over conventional HMM-with-duration.

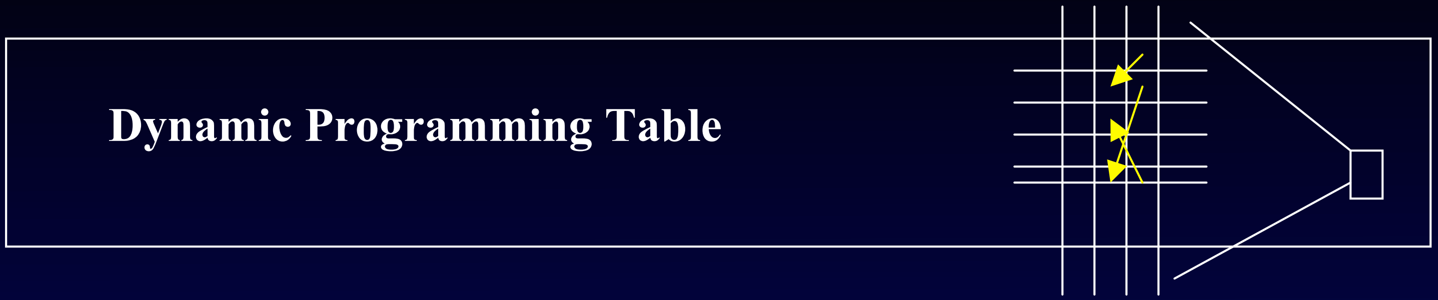
Footprint State & Transition Enumeration

- Atomic states involving exon, e_x, and intron, i_x, are context-specific (frame, direction), but not junk, j.
- Use ê, î in order to denote reverse encodings.
- * Support the 3 stop codons {TAA, TAG, TGA} explicitly.
- The result is 33 allowed transition states …
  - 13 XX-types: $i_x i_x$(3), $\hat{\imath}_x \hat{\imath}_x$(3), $e_x e_y$(3), $\hat{e}_x \hat{e}_y$(3), jj(1)
  - 20 eij-types: $e_x i_x$(3), $\hat{e}_x \hat{\imath}_x$(3), $i_x e_x$(3), $\hat{\imath}_x \hat{e}_y$(3), $e_2 j$(3*), $\hat{e}_2 j$(3*), $j e_0$(1), $j \hat{e}_0$(1)
- Impose minimum length duration on states
- For each eij-dimer generate F-1 footprint states.
- The result is (13+20*(F-1)) allowed footprint states
  - 13 XX-types: $i_x i_x$->$i_x i_x$(3), $\hat{\imath}_x \hat{\imath}_x$->$\hat{\imath}_x \hat{\imath}_x$(3), $e_x e_y$->e_y e_z (3), $\hat{e}_x \hat{e}_y$->$\hat{e}_y \hat{e}_z$(3), jj-> jj(1)
  - 20*(F-1) eij-types: $e_x i_x$(3*(F-1)), $\hat{e}_x \hat{\imath}_x$(3*(F-1)), $i_x e_x$(3*(F-1)), $\hat{\imath}_x \hat{e}_y$(3*(F-1)), $e_2 j$(3*(F-1)), $\hat{e}_2 j$(3 *(F-1)), $j e_0$(1*(F-1)), $j \hat{e}_0$(1*(F-1))
  - Have approx. 20*F states, with typical F=50 footprint sizes will expect to have approx. 1000-state HMM processing via this approach.

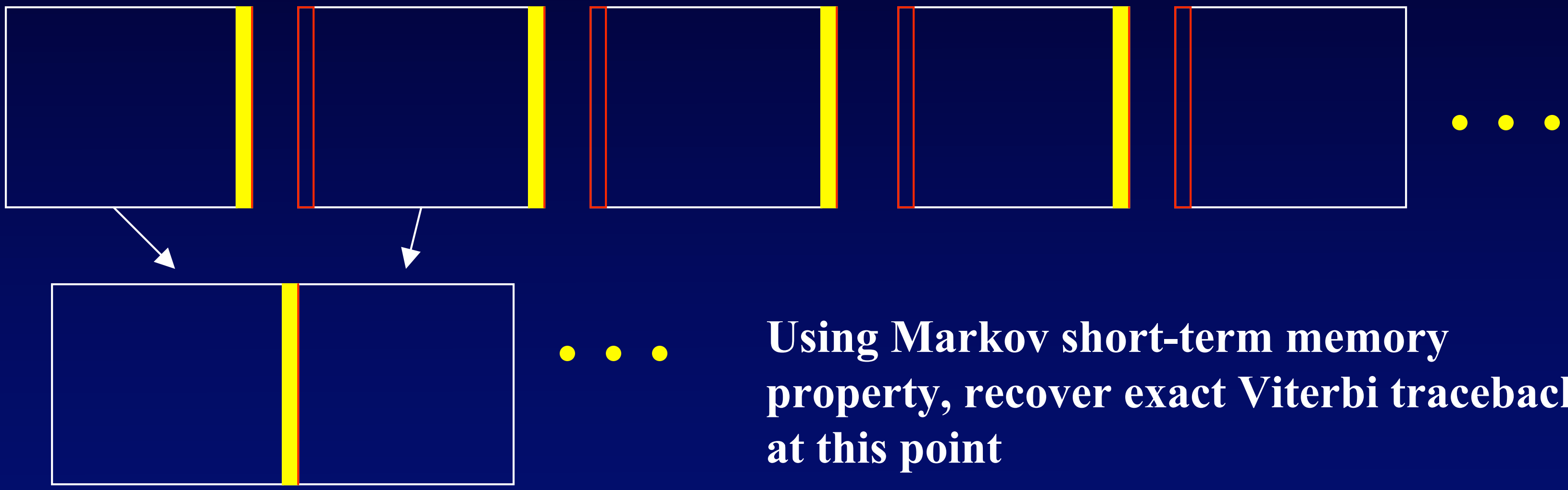
Preliminary Results

- Chromosome I; *C. Elegans*; R=0, L=2, *r*=*l*=3 (or F=6)
  - Individual exon bases
    - sn= 0.88 (matching e base count/ annotated e base count)
    - sp= 0.86 (matching e base count / predicted e base count)
  - Full exon
    - SN= 0.62 (full exon match count / annotated exon count)
    - SP= 0.55 (full exon match count / predicted exon count)
  - Best Overall: 6-6-6-0-vfull_m2_viterbi with 0.70
  - Best je detection: 6-7-6-0-vfull_m2_viterbi with 1.0
  - Best ej detection: 5-5-6-0-vfull_m2_viterbi with 0.78
  - Best ie detection: 1-5-6-0-vfull_m2_viterbi with 0.8125
  - Best ei detection: 2-1-10-0-vfull_m2_viterbi with 0.74

Distributed HMM/EM processing

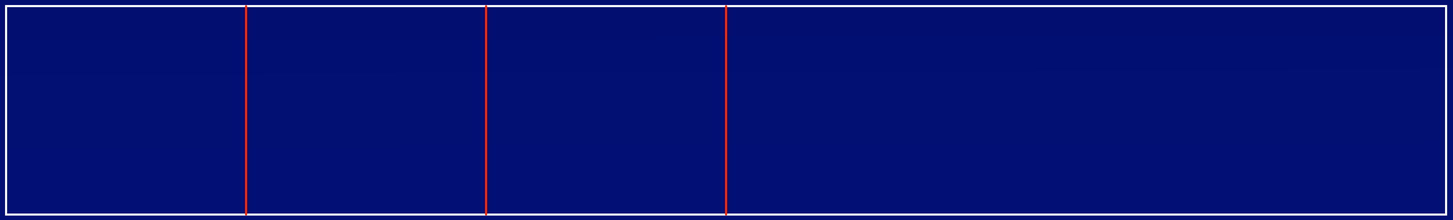


Partitioned Dynamic Programming Table



Using Markov short-term memory property, recover exact Viterbi traceback at this point

Post EM-relaxation of join statistics (recover emission and transition probabilities:



Computational time reduced by ~ N on cluster with N nodes.

Markov Model --> SVM Feature Vector

Markov Model (MM) Profile (*V. cholerae*):

Index: ∞ ….. -17 ……. -2 -1 | 0 1 2 3 …… ∞

------------(A/G)-----------------|(A T G)----------------

A 0.25……..0.4…………..0.4 |.93 0 0 0.4……..0.25

C 0.25……..0.1…………..0.3 |.01 0 0 0.3……..0.25

G 0.25……..0.1…………..0.2 |.60 0 1 0.2……..0.25

T 0.25……..0.4…………..0.1 |.09 1 0 0.1……..0.25

Log odds ratio: $\log[P_{start}(\text{sub-sequence})/P_{non\text{-}start}(\text{sub-sequence})] > 0$ --> a start region

Classifier based on $\log[P_{start}/P_{non\text{-}start}] = \Sigma_i \log[P_{start}(x_i=b_i)/P_{non\text{-}start}(x_i=b_i)]$.

Rather than a classification built on the sum of the independent log odds ratios, the sum of components could be replaced with a vectorization of components:

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

These can be viewed as feature vectors for SVM classification. The SVM partially recovers linkages lost with the Markov short-term memory approx..

UNO Computer Science

Other MM-Variant Algorithms

There are generalizations for the MM sensor, and all are compatible with the SVM f.v. classification profiling.

IMM: the order of the MM is interpolated according to some ***globally*** imposed cut-off criterion, such as a minimum sub-sequence count:

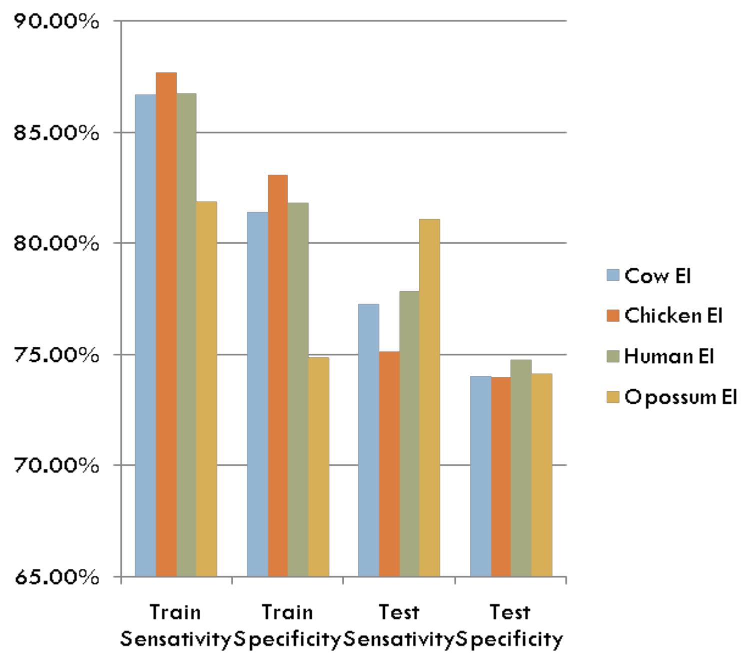
gIMM: like IMM with its count cutoff, but when going to higher order in the interpolation there is no constraint to contiguous sequence elements -- I.e., 'gaps' are allowed. The resolution of what gap-size to choose when going to the next higher order is resolved by evaluating the Mutual Information. Higher orders perform motif analysis as side-effect via sub-sequence correlations to some reference 'scaffolding' (such as the start codon).

hIMM/ghIMM: no longer employ a global cutoff criterion -- count cutoff criterion applied at the sub-sequence level.

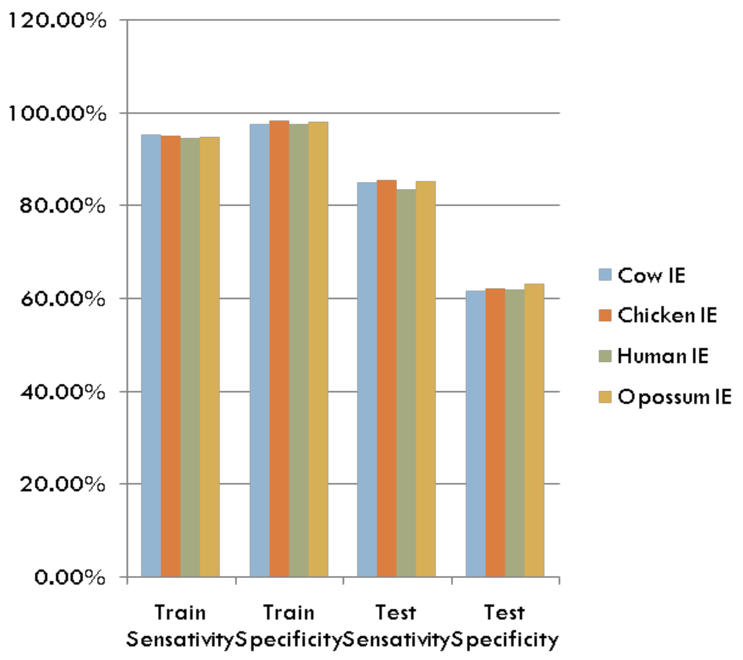
MM, IMM, gIMM, hIMM, ghIMM ==> SVM/MM, SVM/IMM, SVM/gIMM, etc.

SVM Radial Gamma Kernel

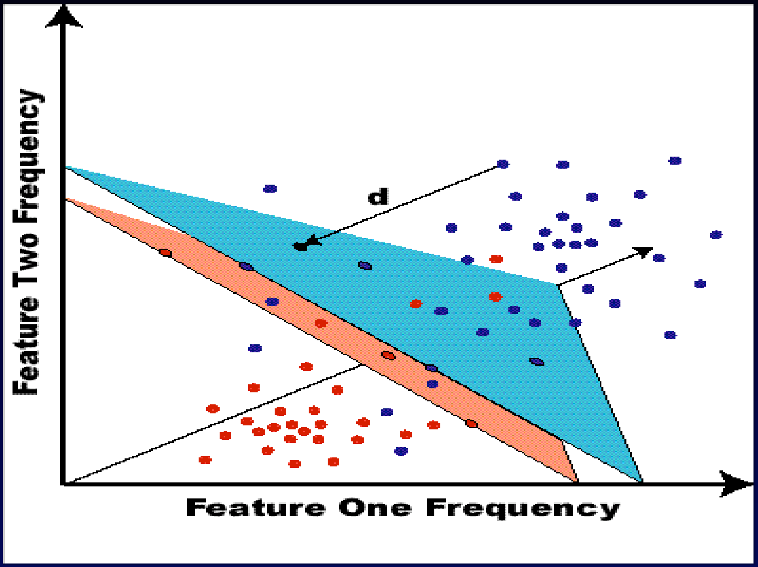
EI Splice Site



IE Splice Site



SVM Discrimination



SVM Implementation	Kernel	Accuracy (100%*(SN+SP)/2)
W-H SMO	*Absdiff*	*94.0*
W-H SMO	*Entropic*	*94.0*
W-H SMO	Gaussian	92.5
Platt SMO	Absdiff	86.5
Platt SMO	Entropic	70.0
Platt SMO	Gaussian	73.5
Keerthi1 SMO	*Absdiff*	*94.0*
Keerthi1 SMO	Entropic	89.5
Keerthi1 SMO	Gaussian	91.5
Keerthi2 SMO	*Absdiff*	*94.0*
Keerthi2 SMO	Entropic	89.5
Keerthi2 SMO	Gaussian	91.5

SVM discrimination is so strong and stable, and user friendly, that it may serve a fundamental building-block role in Machine Learning like Integrated Circuit components in Circuit Design

Kernel Model Fitting:

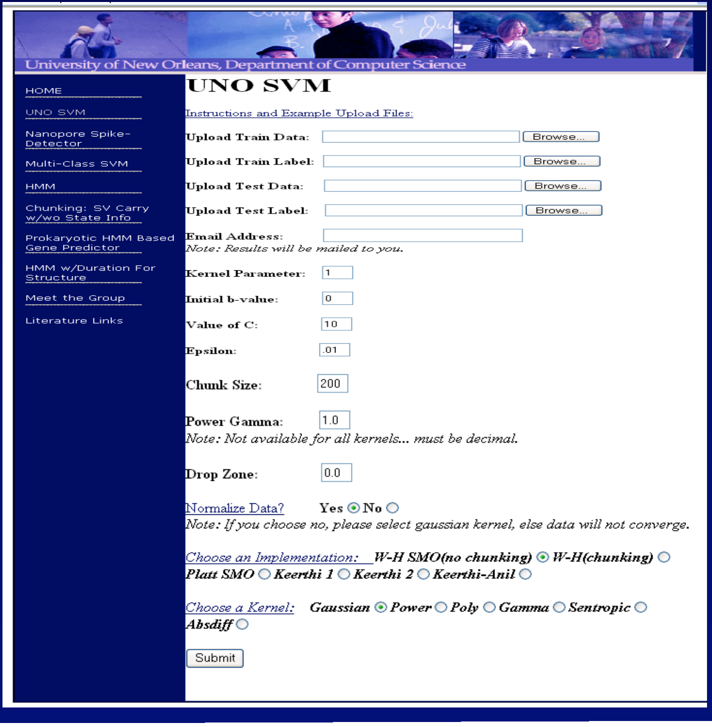
Distance-based Kernels (geometric):

-- $d^2(x,y)=\Sigma_k(x_k-y_k)^2$ (Gaussian);

-- $d^2=(\Sigma_k|x_k-y_k|)^{1/2}$ (Absdiff);

Divergence-based Kernels (Entropic):

-- $d^2(x,y)=D(x||y)+D(y||x)$


University of New Orleans, Department of Computer Science
HOME
UNO SVM
Nanopore Spike-Detector
Multi-Class SVM
HMM
Chunking: SV Carry w/wo State Info
Prokaryotic HMM Based Gene Predictor
HMM w/Duration For Structure
Meet the Group
Literature Links
Browse...
Submit

UNO SVM

Instructions and Example Upload Files:

Upload Train Data:

Upload Train Label:

Upload Test Data:

Upload Test Label:

Email Address:
Note: Results will be mailed to you.

Kernel Parameter: 1

Initial b-value: 0

Value of C: 10

Epsilon: .01

Chunk Size: 200

Power Gamma: 1.0
Note: Not available for all kernels... must be decimal.

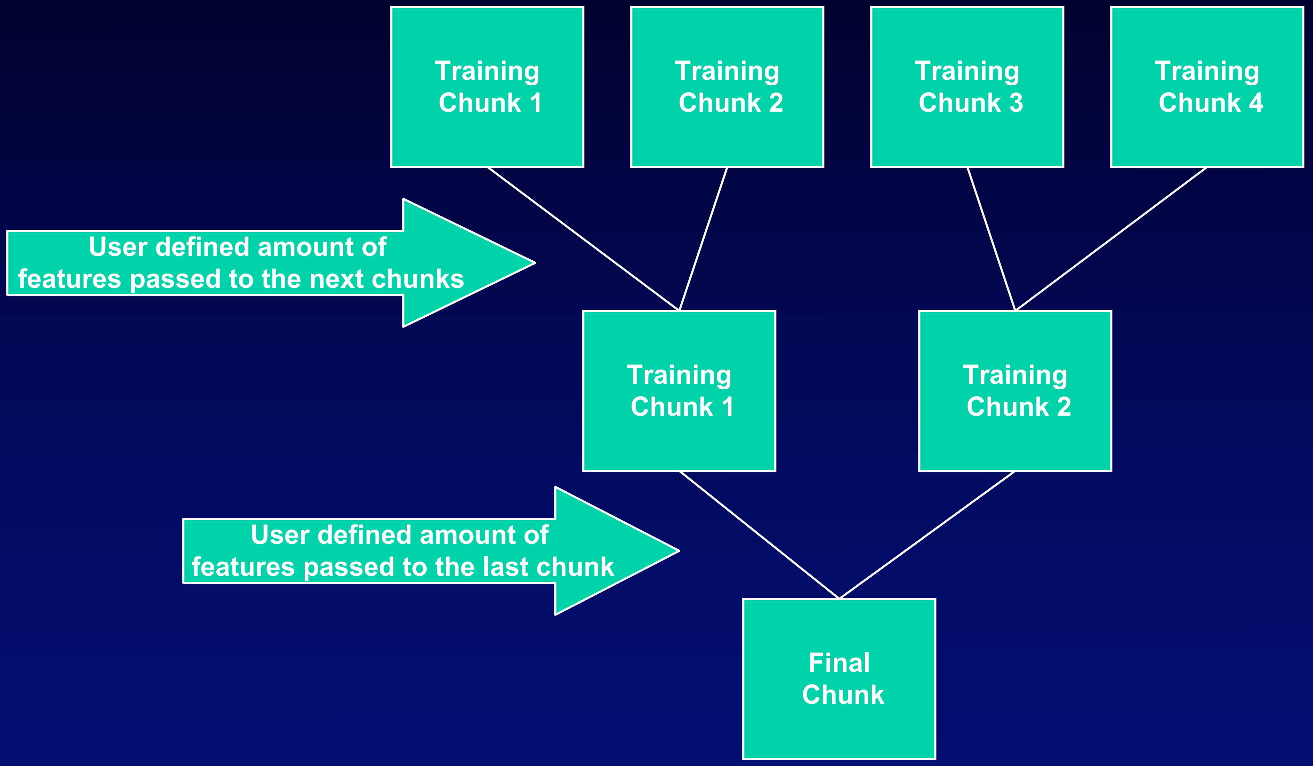
Drop Zone: 0.0

Normalize Data? [✓ Yes] No
Note: If you choose no, please select gaussian kernel, else data will not converge.

Choose an Implementation: [✓ ***W-H SMO(no chunking)***] ***W-H(chunking)*** ***Platt SMO*** ***Keerthi 1*** ***Keerthi 2*** ***Keerthi-Anil***

Choose a Kernel: [✓ ***Gaussian***] ***Power*** ***Poly*** ***Gamma*** ***Sentropic*** ***Absdiff***

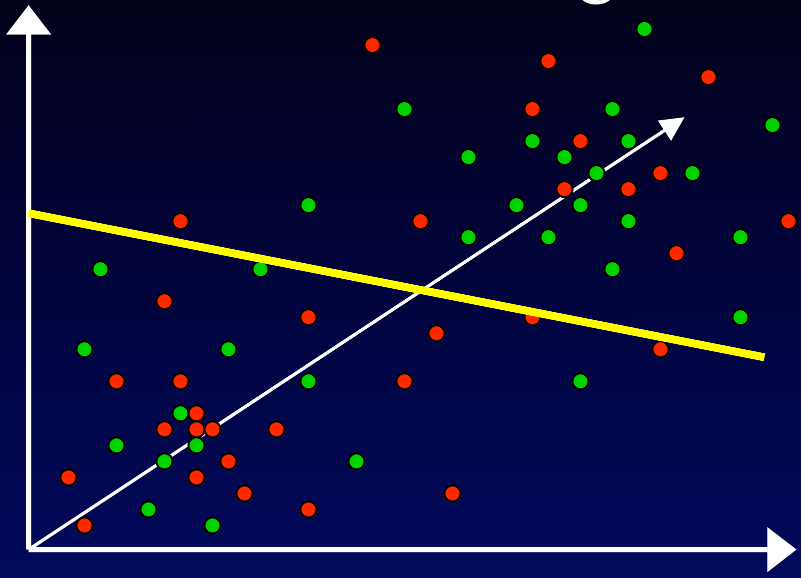
Distributed Chunking


Training Chunk 1
Training Chunk 2
Training Chunk 3
Training Chunk 4
User defined amount of features passed to the next chunks
Training Chunk 1
Training Chunk 2
User defined amount of features passed to the last chunk
Final Chunk

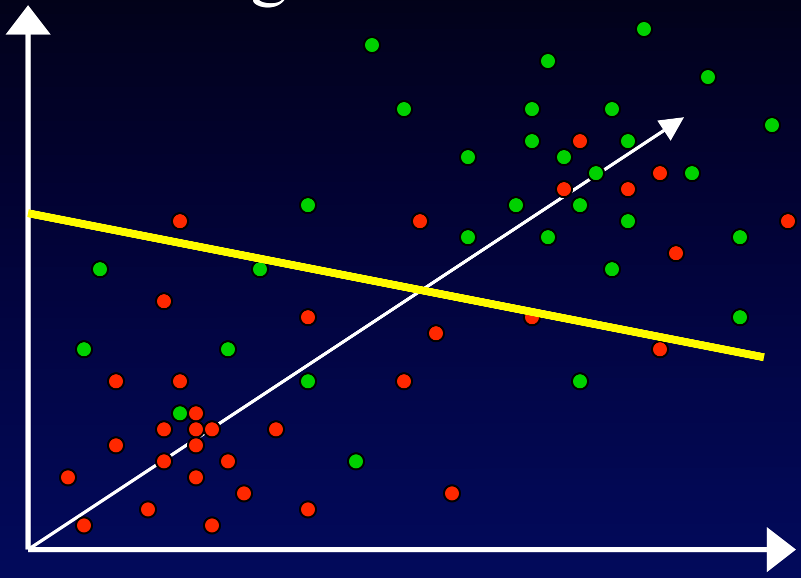
speed up of [illegible]

SVM-based Clustering (via multi-pass SVM)

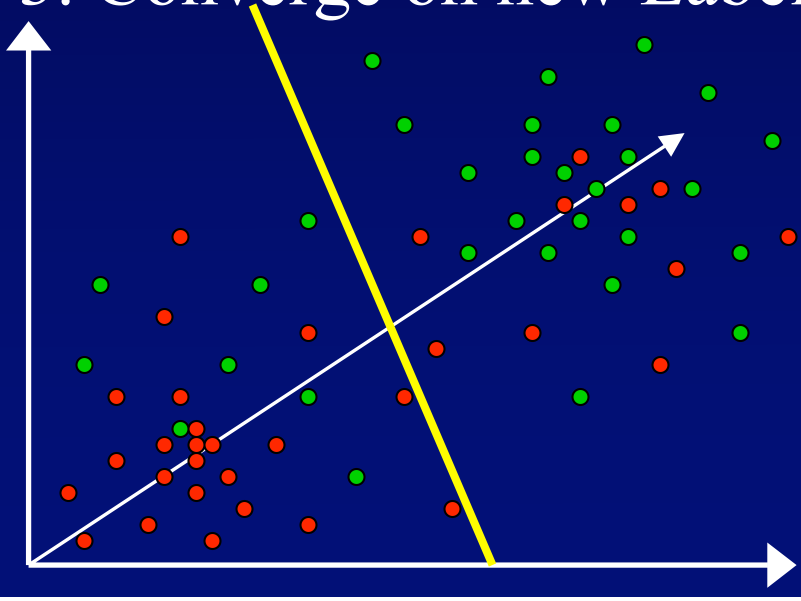
1. Label & Converge:



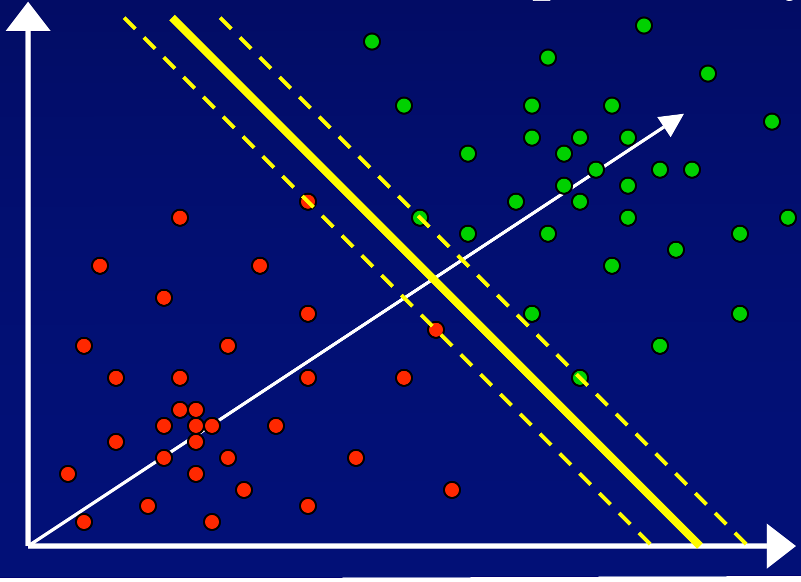
2. Change Weakest Labels:



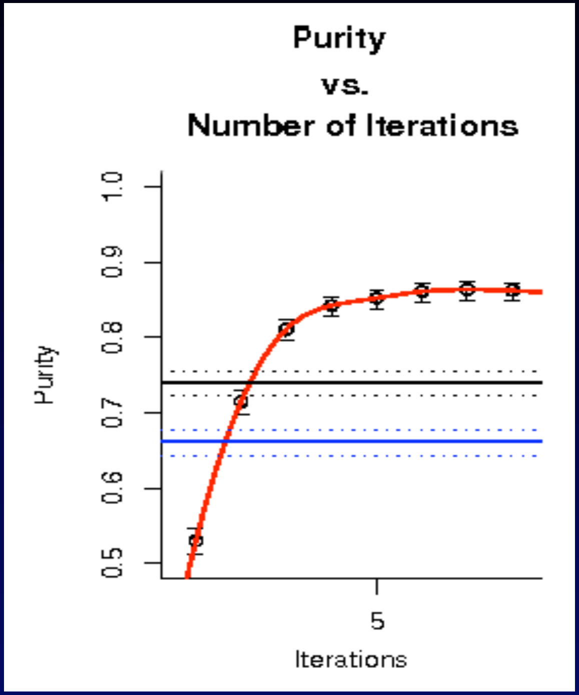
3. Converge on new Labels:

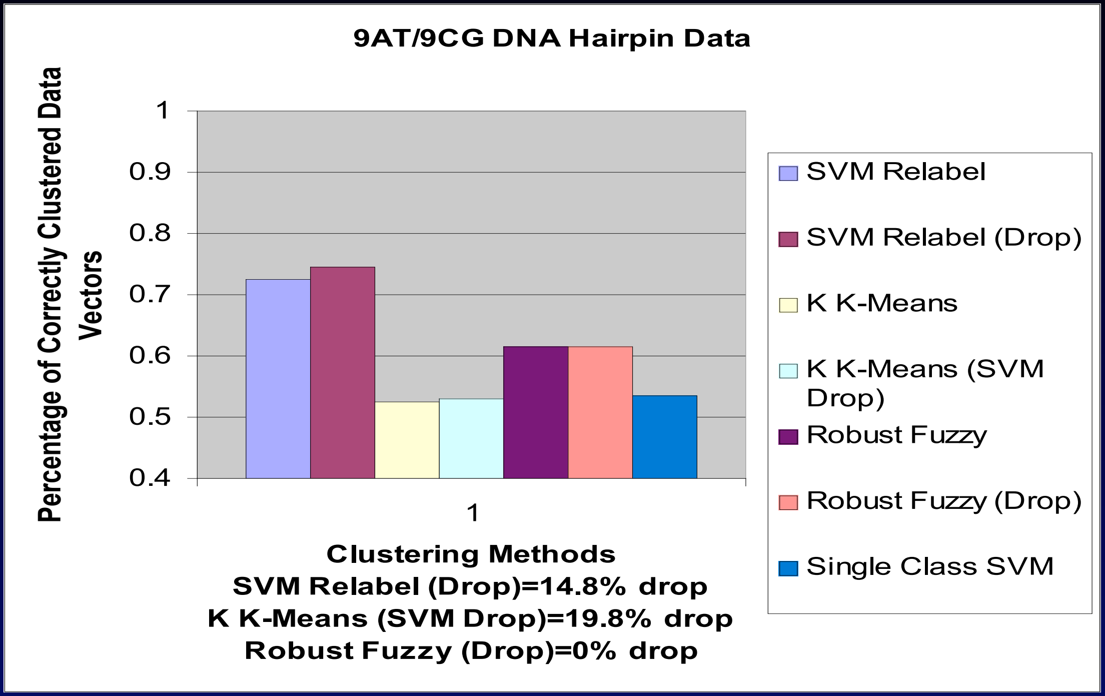


4. Iterate until Separability:

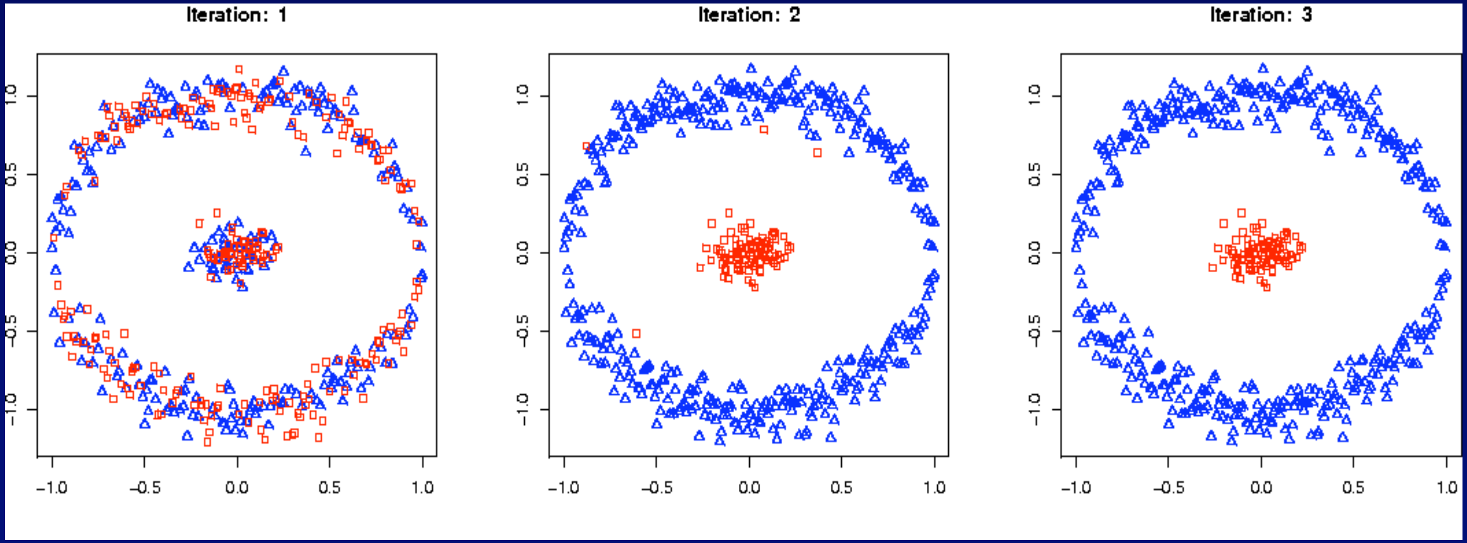


SVM-based Clustering outperforms other methods





SVM Relabel (Drop)=14.8% drop
K K-Means (SVM Drop)=19.8% drop
Robust Fuzzy (Drop)=0% drop


Iteration: 1
Iteration: 2
Iteration: 3