Accurate and Interpretable Large Scale Genomic Signal Detection

(Support Vector Machine Based Signal Detectors)

Sören Sonnenburg
Friedrich Miescher Laboratory, Tübingen

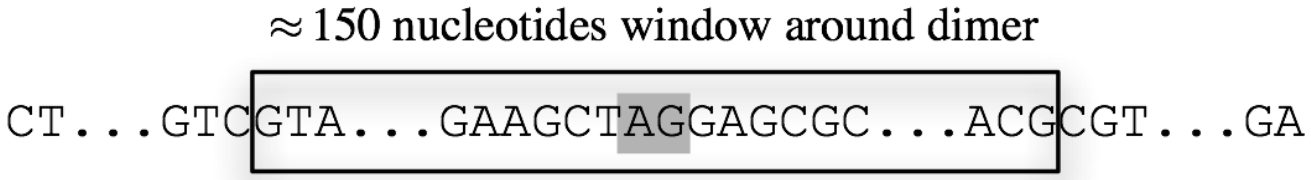
joint work with
Alexander Zien, Jonas Behr, Gabriele Schweikert, Petra Philips and Gunnar Rätsch

Outline

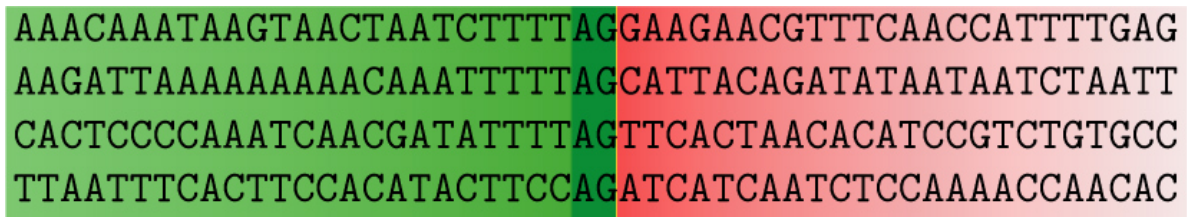
Outline

Recognizing Genomic Signals

Discriminate true signal positions against all other positions



- True sites: fixed window around a true site
- Decoy sites: all other consensus sites



Examples: Transcription start site finding, splice site prediction, alternative splicing prediction, trans-splicing, polyA signal detection, translation initiation site detection

Types of Signal Detection Problems I

Vague categorization

(based on positional variability of motifs)

Position Independent

→ Motifs may occur anywhere,

```
AAACAAAAACGTAACTAATCTTTTAGAGAGAACGTTTCAACCATTTTGAG
AAGATTAACTCATCACAGATTTCATTACATACAGATATAATTCAAAAATT
CACTCCCCAAATCAACGATATTTAAAAATCACTAACACATCCGTCTGTGC
```

e.g. tissue classification using promotor region

Types of Signal Detection Problems II

Vague categorization

(based on positional variability of motifs)

Position Dependent

→ Motifs very stiff, almost always at same position,

```
AAACAAATAAGTAACTAATCTTTTAAGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAACATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAATTCACTAACACATCCGTCTGTGCC
```

e.g. Splice Site Classification

Types of Signal Detection Problems III

Vague categorization

(based on positional variability of motifs)

Mixture Position Dependent/Independent

→ variable but still positional information

```
AAACAAATAAGTAACTAATCTTTTAAAGAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTCATTAAATACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTAAATTTCACTAACACATCCGTCTGTGC
```

e.g. Promoter Classification

Classification - Learning based on examples

Given:

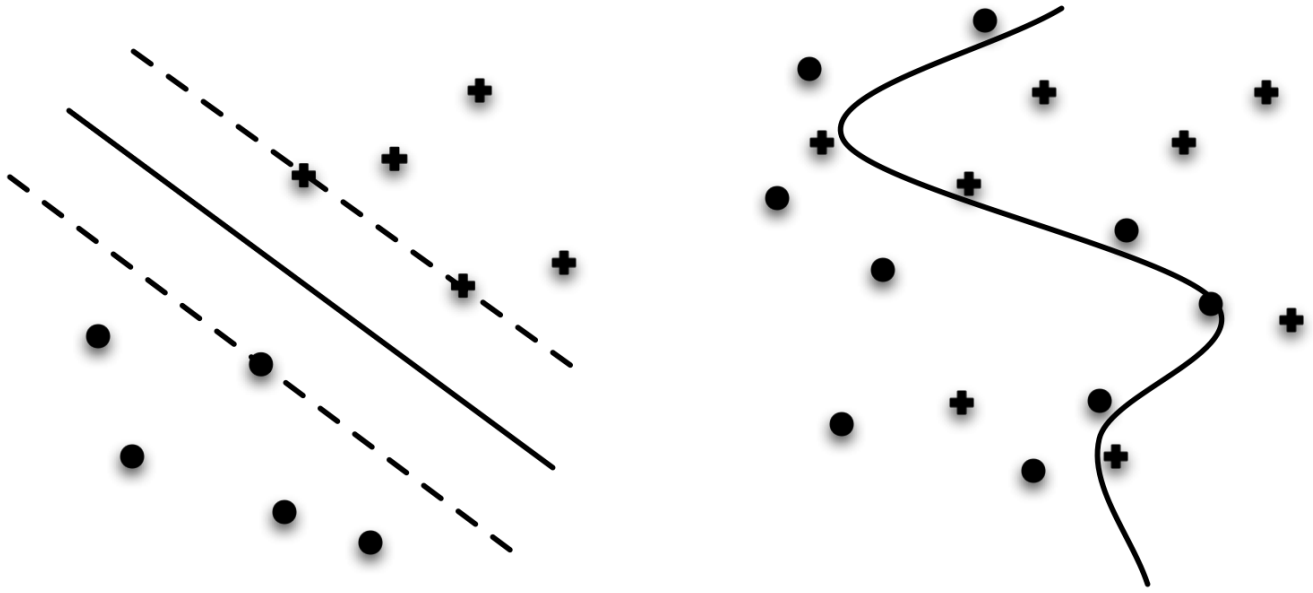
Training examples $(\mathbf{x}_i, y_i)_{i=1}^{N} \in (\{A, C, G, T\}^L, \{-1, +1\})^N$

```
AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC
TTGTTTTAATATTCAATTTTTTACAGTAAGTTGCCAATTCAATGTTCCAC
TACCTAATTATGAAATTAAAATTCAGTGTGCTGATGGAAACGGAGAAGTC
```

Wanted:

Function (Classifier) $f(\mathbf{x}) : \{A, C, G, T\}^L \mapsto \{-1, +1\}$

Support Vector Machines (SVMs)



- **Support Vector Machines** learn weights $\alpha \in \mathbb{R}^N$ over training examples in kernel feature space $\Phi : \mathbf{x} \mapsto \mathbb{R}^D$,

$$f(\mathbf{x}) = \operatorname{sign}\left(\sum_{i=1}^{N} y_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b\right),$$

with kernel $k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$

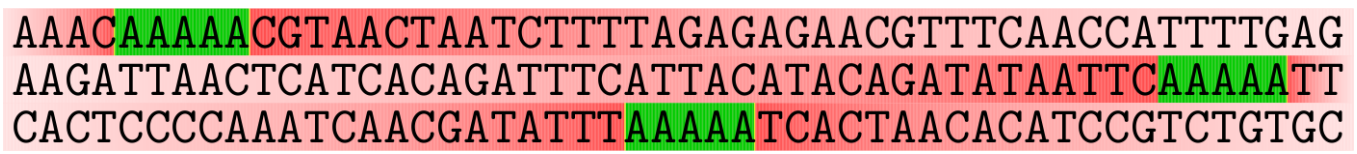
The Spectrum Kernel

Support Vector Machine

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i \mathsf{k}(\mathbf{x}, \mathbf{x}_i) + b\right),$$

Spectrum Kernel (with mismatches, gaps)

$$K(\mathbf{x}, \mathbf{x}') = \Phi_{sp}(\mathbf{x}) \cdot \Phi_{sp}(\mathbf{x}')$$

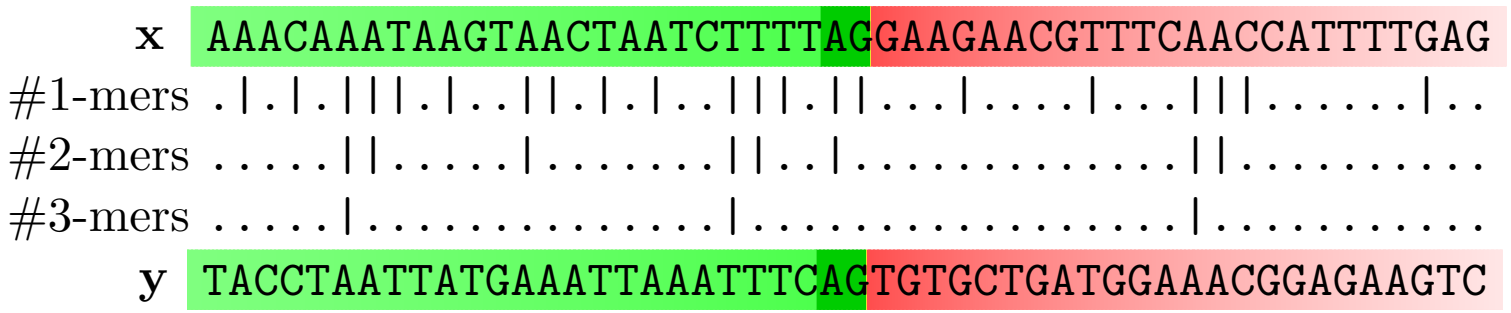


The Weighted Degree Kernel

Support Vector Machine

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i \mathrm{k}(\mathbf{x}, \mathbf{x}_i) + b\right),$$

$$\mathrm{k}(\mathbf{x}, \mathbf{x}') = \sum_{k=1}^{K} \beta_k \sum_{i=1}^{L-k+1} \mathbb{I}\left\{\mathbf{x}[i]^k = \mathbf{x}'[i]^k\right\}.$$

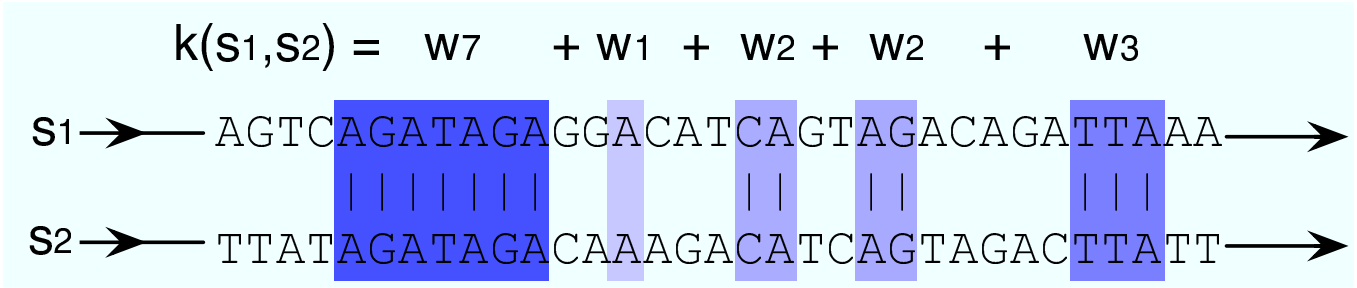


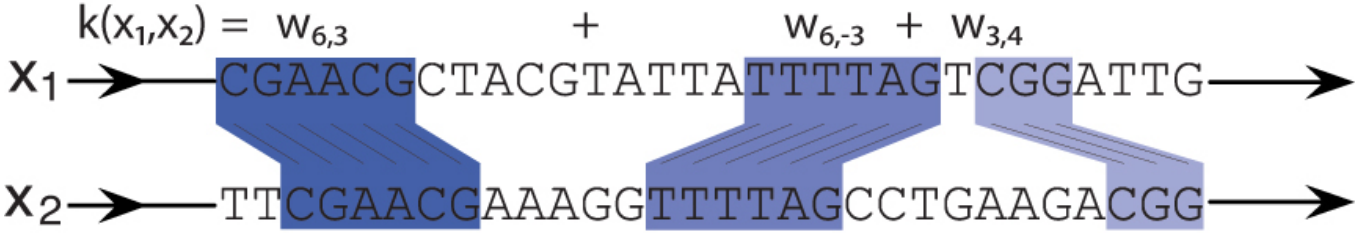
Example: $K = 3$: $\mathrm{k}(\mathbf{x}, \mathbf{x}') = \beta_1 \cdot 21 + \beta_2 \cdot 8 + \beta_3 \cdot 3$

The Weighted Degree Kernel with *shifts*

Support Vector Machine

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b\right),$$





Accelerating String-Kernel-SVMs

1. Linear run-time of the kernel
2. Accelerating linear combinations of kernels

Idea of the `Linadd` Algorithm:

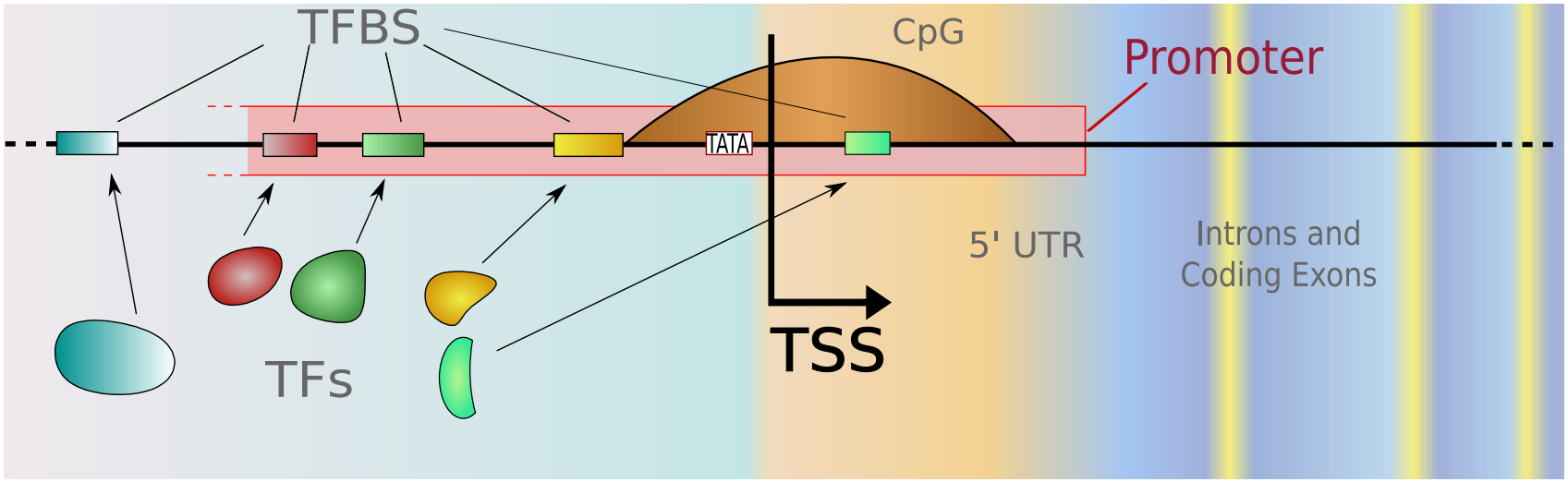
Store $\mathbf{w}$ and compute $\mathbf{w} \cdot \Phi(\mathbf{x})$ *efficiently*

$$f(\mathbf{x}_j) = \sum_{i=1}^{N} \alpha_i y_i \, \mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \underbrace{\sum_{i=1}^{N} \alpha_i y_i \Phi(\mathbf{x}_i)}_{\mathbf{w}} \cdot \Phi(\mathbf{x}_j) = \mathbf{w} \cdot \Phi(\mathbf{x}_j)$$

Possible for low-dimensional or sparse $\mathbf{w}$

Effort: $\mathcal{O}(ML) \Rightarrow$ **speedup of factor** N

Detecting Transcription Start Sites



- POL II binds to a rather vague region of $\approx [-20, +20]$ bp
- Upstream of TSS: promoter containing transcription factor binding sites
- Downstream of TSS: 5' UTR, and further downstream coding regions and introns (different statistics)
- 3D structure of the promoter must allow the transcription factors to bind

$\Rightarrow$ **Promoter Prediction is non-trivial**

Features to describe the TSS

- TFBS in Promoter region
- condition: DNA should not be too twisted
- CpG islands (often over TSS/first exon; in most, but not all promoters)
- TSS with TATA box (≈ -30 bp upstream)
- Exon content in UTR 5" region
- Distance to first donor splice site

Idea:

Combine weak features to build strong promoter predictor

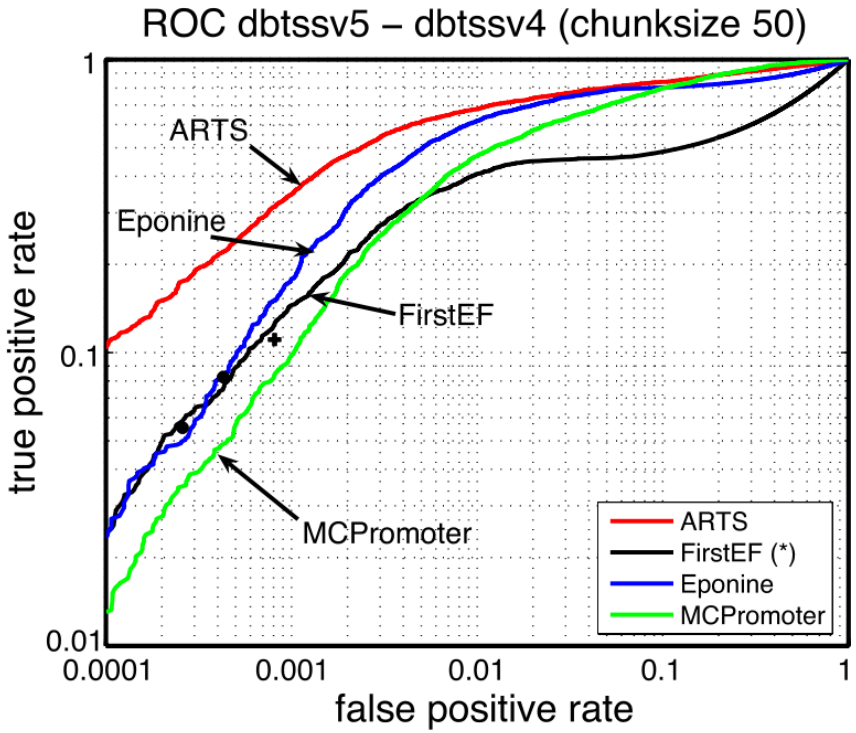
$$k(\mathbf{x}, \mathbf{x}') = k_{TSS}(\mathbf{x}, \mathbf{x}') + k_{CpG}(\mathbf{x}, \mathbf{x}') + k_{coding}(\mathbf{x}, \mathbf{x}') + k_{energy}(\mathbf{x}, \mathbf{x}') + k_{twist}(\mathbf{x}, \mathbf{x}')$$

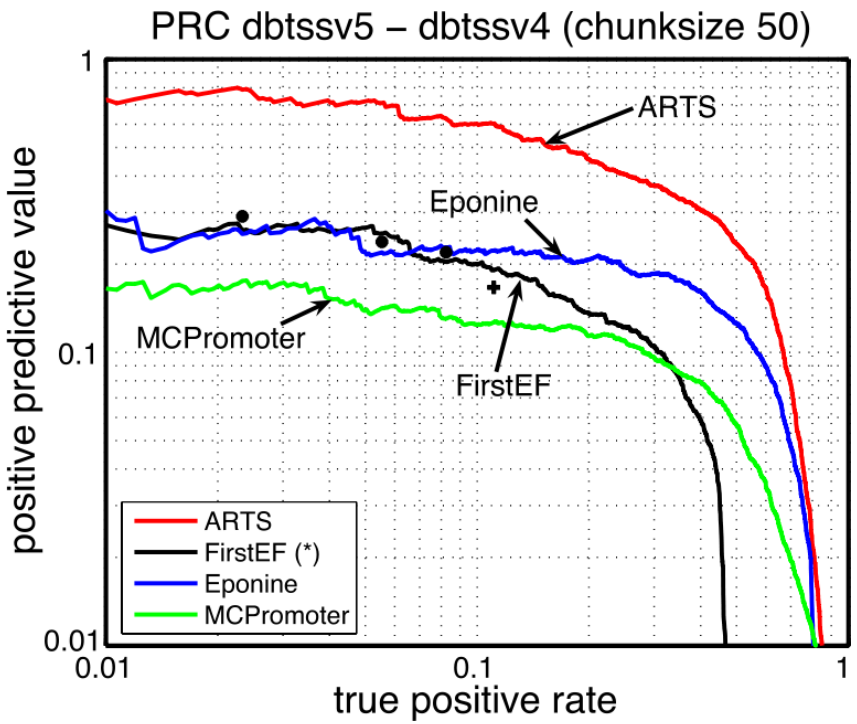
The 5 sub-kernels

1. TSS signal (including parts of core promoter with TATA box)
   – use **Weighted Degree Shift kernel**
2. CpG Islands, distant enhancers and TFBS upstream of TSS
   – use **Spectrum kernel** (large window upstream of TSS)
3. Model coding sequence TFBS downstream of TSS
   – use another **Spectrum kernel** (small window downstream of TSS)
4. Stacking energy of DNA
   – use *btwist* energy of dinucleotides with **Linear kernel**
5. Twistedness of DNA
   – use *btwist* angle of dinucleotides with **Linear kernel**

State-of-the-art Performance

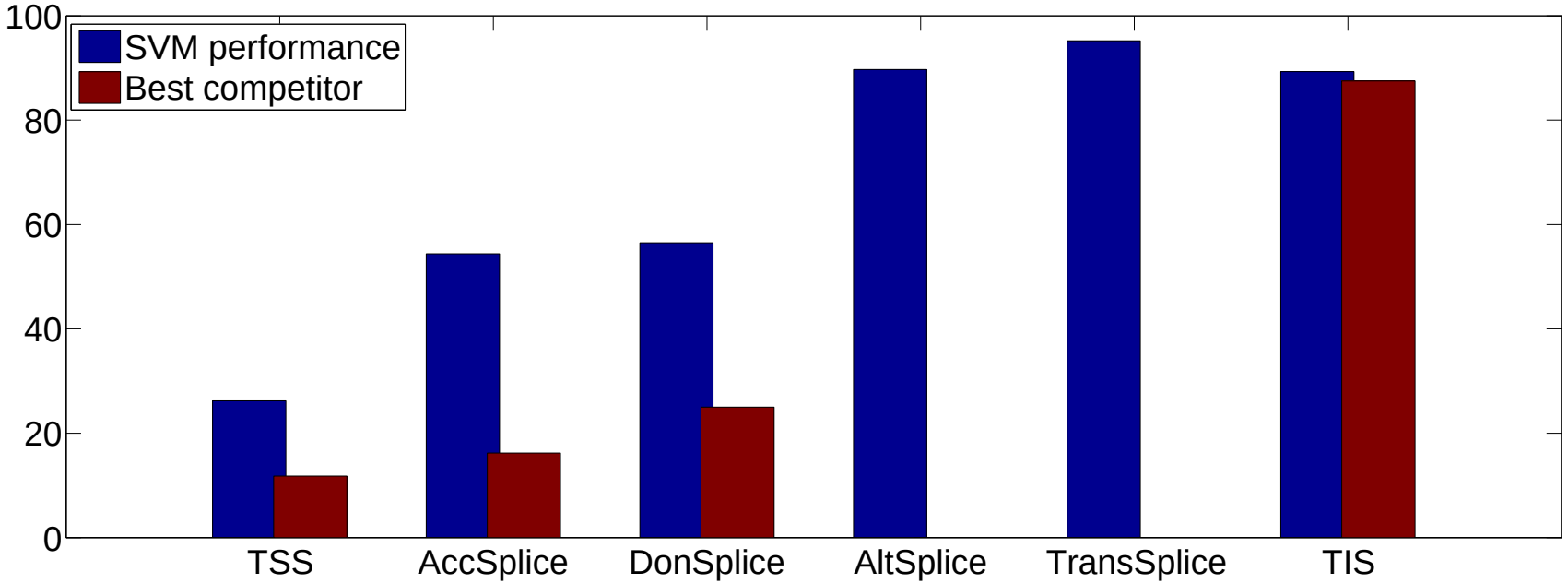
Receiver Operator Characteristic Curve and Precision Recall Curve





⇒ 35% **true positives at a false positive rate of** 1/1000 **(best other method find about a half (**18%**))**

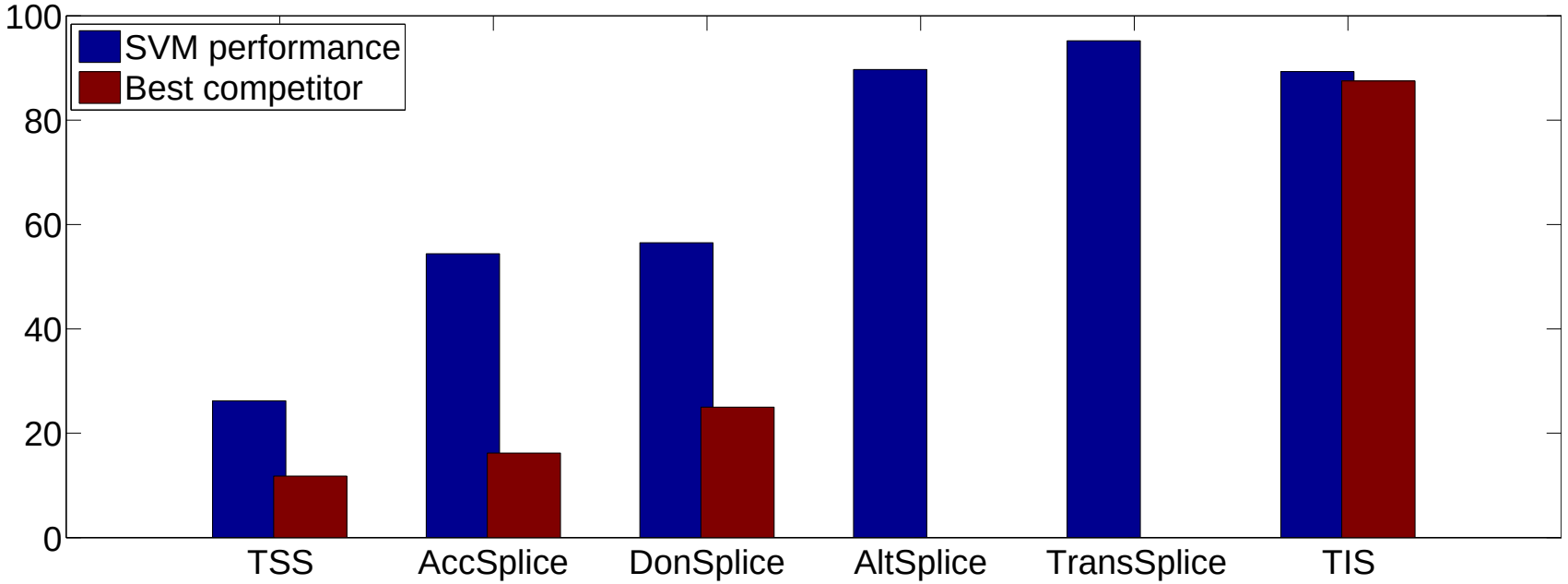
Beauty in Generality



- Transcription Start (Sonnenburg et al., Eponine Down et al.)
- Acceptor Splice Site (Sonnenburg et al.)
- Donor Splice Site (Sonnenburg et al.)
- Alternative Splicing (Rätsch et al., -)
- Transsplicing (Schweikert et al., -)
- Translation Initiation (Sonnenburg et al., Saeys et al.)

Drawback: SVM solution is hard to interpret!!

Beauty in Generality



- Transcription Start (Sonnenburg et al., Eponine Down et al.)
- Acceptor Splice Site (Sonnenburg et al.)
- Donor Splice Site (Sonnenburg et al.)
- Alternative Splicing (Rätsch et al., -)
- Transsplicing (Schweikert et al., -)
- Translation Initiation (Sonnenburg et al., Saeys et al.)

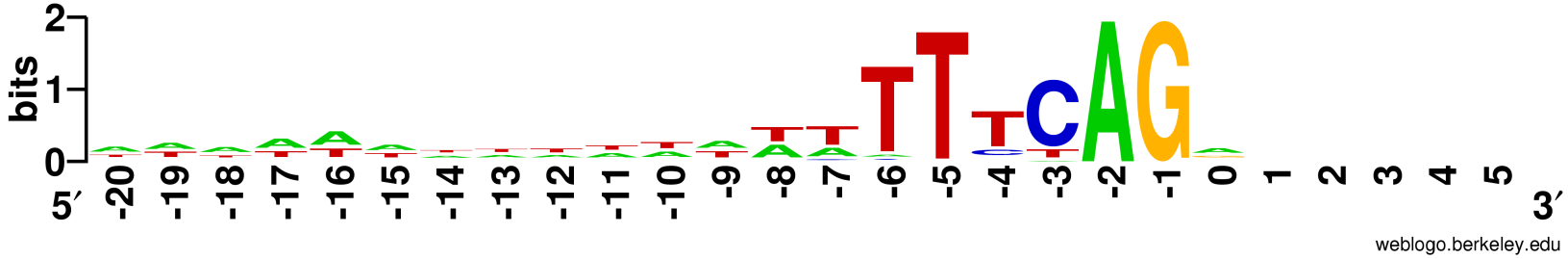
Drawback: SVM solution is hard to interpret!!

Outline

Understanding Support Vector Machines

Goal

For PWMs we have sequence logos:

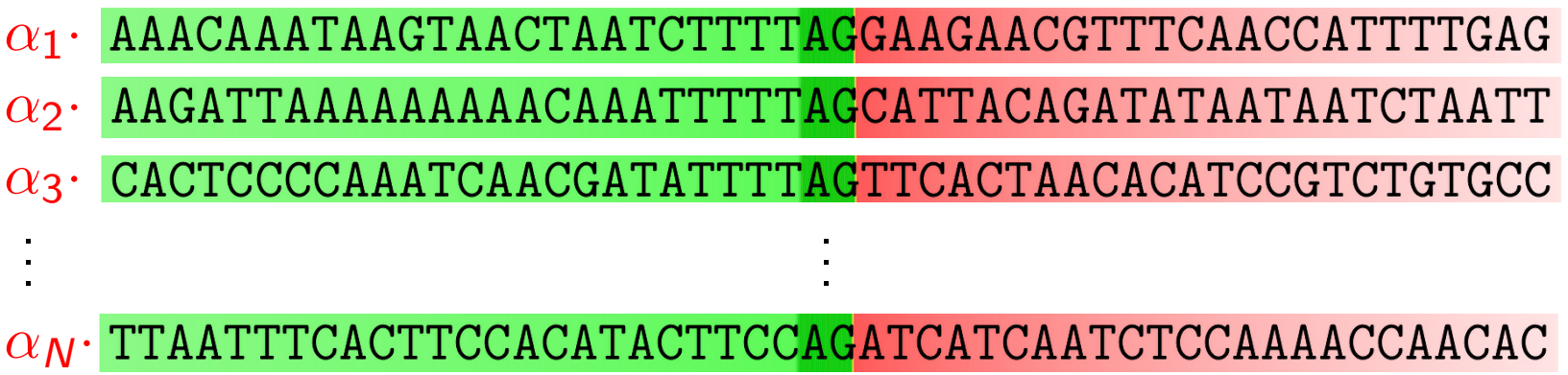


We would like to have **similar means to understand Support Vector Machines.**

Why Are SVM's Hard to Interpret?

SVM decision function is α weighting of training points

$$s(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i \, \mathrm{k}(\mathbf{x}_i, \mathbf{x}) + b$$



But we are interested in weights over features.

SVM Scoring Function

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \Phi(\mathbf{x}_i)$$

$$s(\mathbf{x}) := \sum_{k=1}^{K} \sum_{i=1}^{L-k+1} w\left(\mathbf{x}[i]^k, i\right) + b$$

k-mer	pos. 1	pos. 2	pos. 3	pos. 4	⋯
A	+0.1	-0.3	-0.2	+0.2	⋯
C	0.0	-0.1	+2.4	-0.2	⋯
G	+0.1	-0.7	0.0	-0.5	⋯
T	-0.2	-0.2	0.1	+0.5	⋯
AA	+0.1	-0.3	+0.1	0.0	⋯
AC	+0.2	0.0	-0.2	+0.2	⋯
⋮	⋮	⋮	⋮	⋮	⋱
TT	0.0	-0.1	+1.7	-0.2	⋯
AAA	+0.1	0.0	0.0	+0.1	⋯
AAC	0.0	-0.1	+1.2	-0.2	⋯
⋮	⋮	⋮	⋮	⋮	⋱
TTT	+0.2	-0.7	0.0	0.0	⋯

The Scoring System - Examples

$$s(\mathbf{x}) := \sum_{k=1}^{K} \sum_{i=1}^{L-k+1} w\left(\mathbf{x}[i]^k, i\right) + b$$

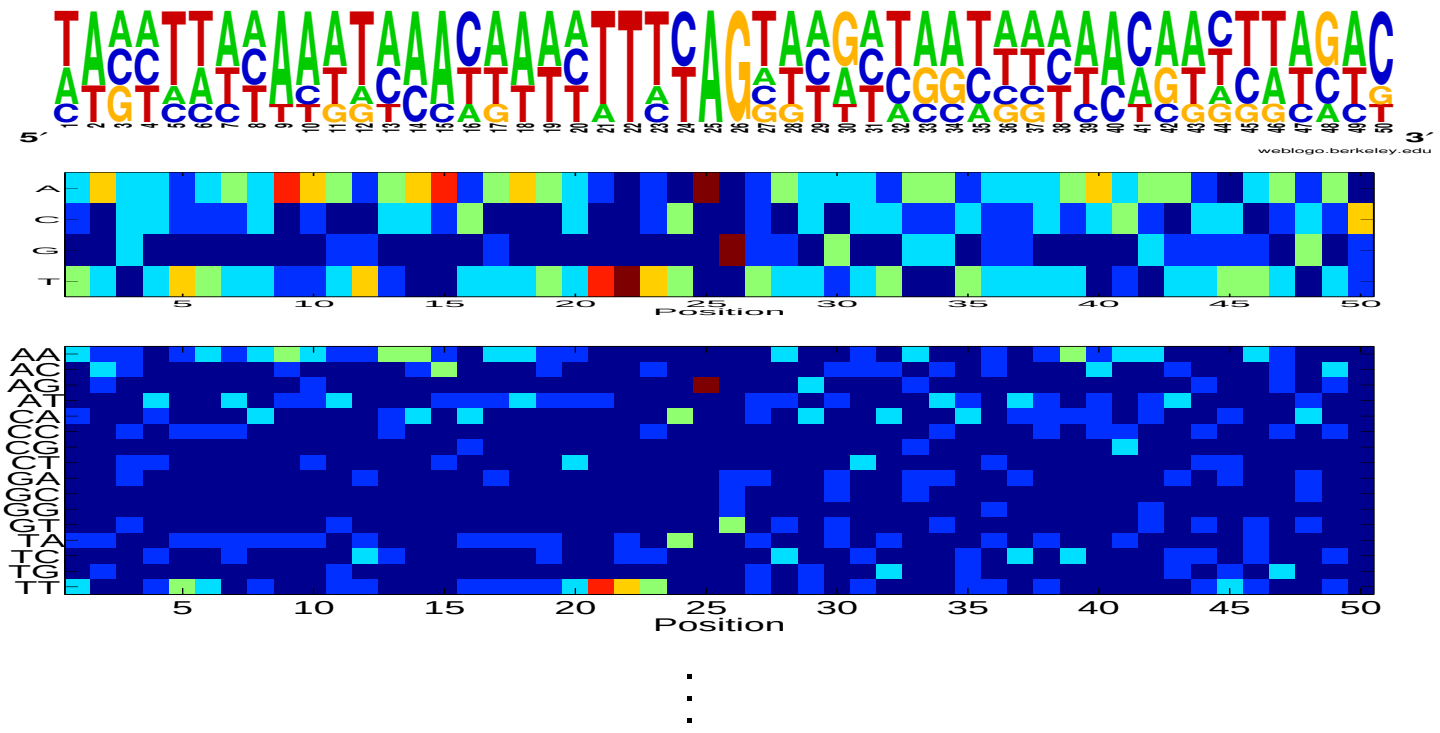
Examples:

- WD-kernel (Rätsch, Sonnenburg, 2005)
- WD-kernel with shifts (Rätsch, Sonnenburg, 2005)
- Spectrum kernel (Leslie, Eskin, Noble, 2002)
- Oligo Kernel (Meinicke et al., 2004)

Not limited to SVM's:

- Markov Chains (higher order/inhomogeneous/mixed order)

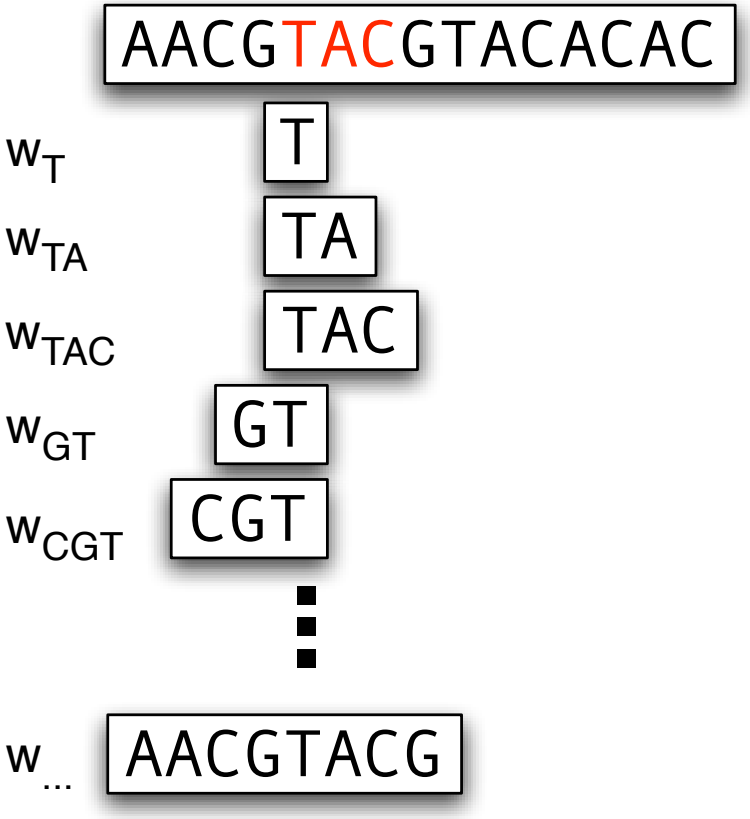
The SVM Weight Vector **w**



- Explicit representation of **w** allows for (some) interpretation!
- String kernel SVMs capable of efficiently dealing with large k-mers $k > 10$

But: Weights for substrings not independent

Interdependence of $k-$mer Weights



What is the score for TAC?

- Take w_{TAC}?
- But substrings and overlapping strings contribute too!

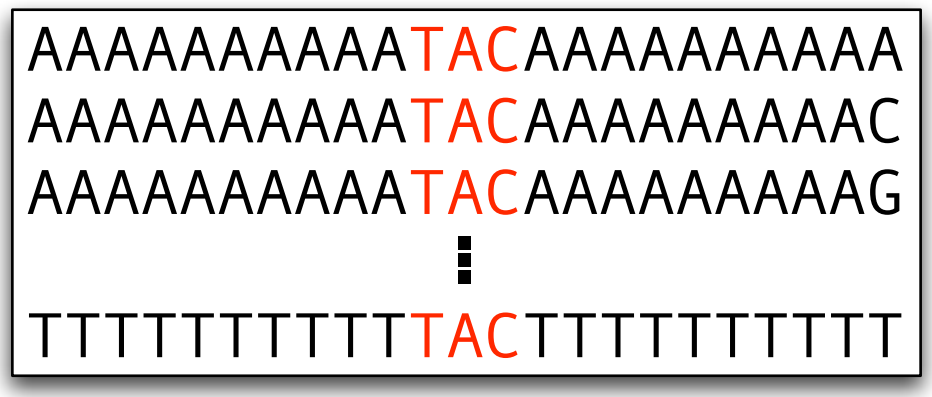
Problem

The SVM-w does NOT reflect the score for a motif

Positional Oligomer Importance Matrices (POIMs)

Idea:

- Given $k-$mer **z** at position j in the sequence, compute expected score $\mathbb{E}[\,s(\mathbf{x}) \mid \mathbf{x}[j] = \mathbf{z}\,]$ (**for small k**)



- Normalize with *expected score* over **all sequences**

POIMs

$$Q(\mathbf{z}, j) := \mathbb{E}[\,s(\mathbf{x}) \mid \mathbf{x}[j] = \mathbf{z}\,] - \mathbb{E}[\,s(\mathbf{x})\,]$$

$\Rightarrow$ **Needs efficient algorithm for computation**

Efficient Computation

Effort of naive approach exponential $\mathcal{O}(|\Sigma|^L + L|\Sigma|^k)$
(e.g. Splice Sites 10^{120})

$$Q(\mathbf{z}, j) := \mathbb{E}[s(\mathbf{x}) \mid \mathbf{x}[j] = \mathbf{z}] - \mathbb{E}[s(\mathbf{x})]$$

- Number of k-mers grows linearly with size of input
- Only features which are dependent on $(\mathbf{z}, j)$ matter
- Computation can be split in contributions from 4 cases

Efficient Recursive Algorithm:

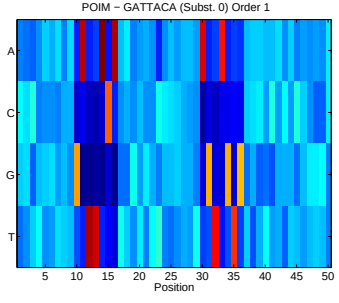
Effort linear in length of input: $\mathcal{O}(LN + L|\Sigma|^k)$

Ranking Features and Condensing Information

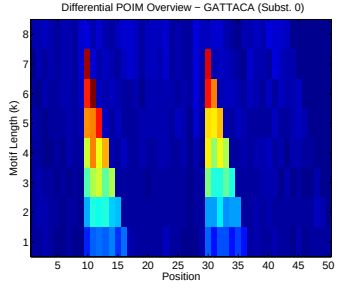
- Obtain highest scoring $\mathbf{z}$ from $Q(\mathbf{z}, i)$ (Enhancer or Silencer)

$\mathbf{z}$	i	$Q(\mathbf{z}, i)$
GATTACA	10	+30
AGTAGTG	30	+20
AAAAAAA	10	-10
...	...	...

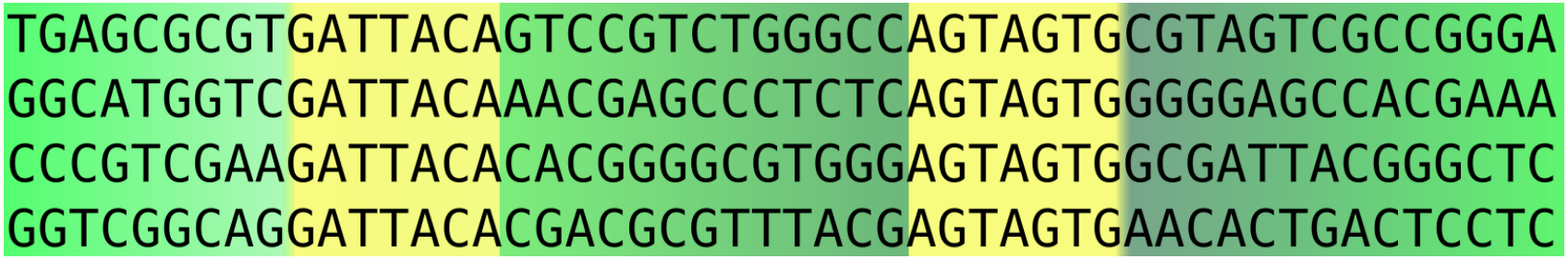
- Visualize POIM as heat map;
  x-axis: position
  y-axis: k-mer
  color: importance



- For large k: Differential POIMs;
  x-axis: position
  y-axis: k-mer length
  color: importance



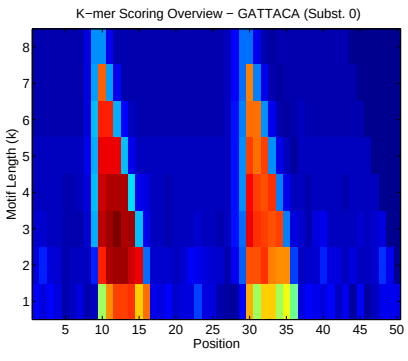
GATTACA and AGTAGTG at Fixed Positions 10 and 30



GATTACA and AGTAGTG at fixed positions 10 and 30

TGAGCGCGTGATTACAGTCCGTCTGGGCCAGTAGTGCGTAGTCGCCGGGA
GGCATGGTCGATTACAAACGAGCCCTCTCAGTAGTGGGGGAGCCACGAAA
CCCGTCGAAGATTACACACGGGGCGTGGGAGTAGTGGCGATTACGGGCTC
GGTCGGCAGGATTACACGACGCGTTTACGAGTAGTGAACACTGACTCCTC

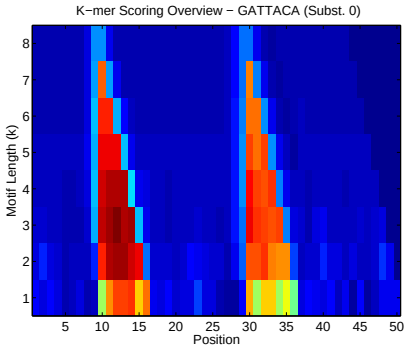
w



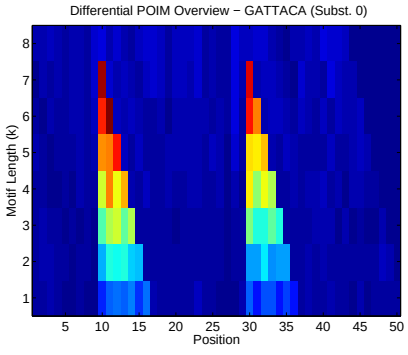
GATTACA and AGTAGTG at fixed positions 10 and 30

TGAGCGCGTGATTACAGTCCGTCTGGGCCAGTAGTGCGTAGTCGCCGGGA
GGCATGGTCGATTACAAACGAGCCCTCTCAGTAGTGGGGGAGCCACGAAA
CCCGTCGAAGATTACACACGGGGCGTGGGAGTAGTGGCGATTACGGGCTC
GGTCGGCAGGATTACACGACGCGTTTACGAGTAGTGAACACTGACTCCTC

w



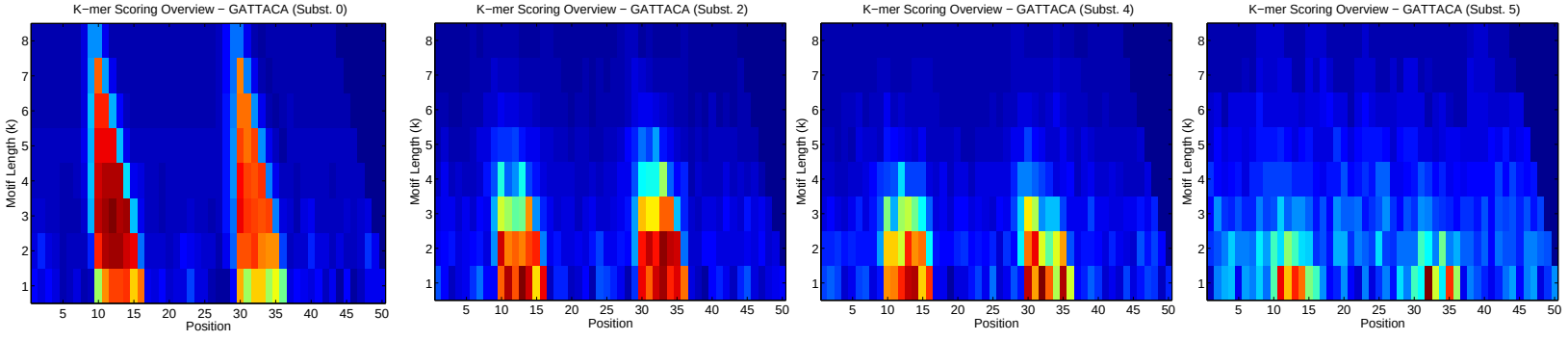
Q



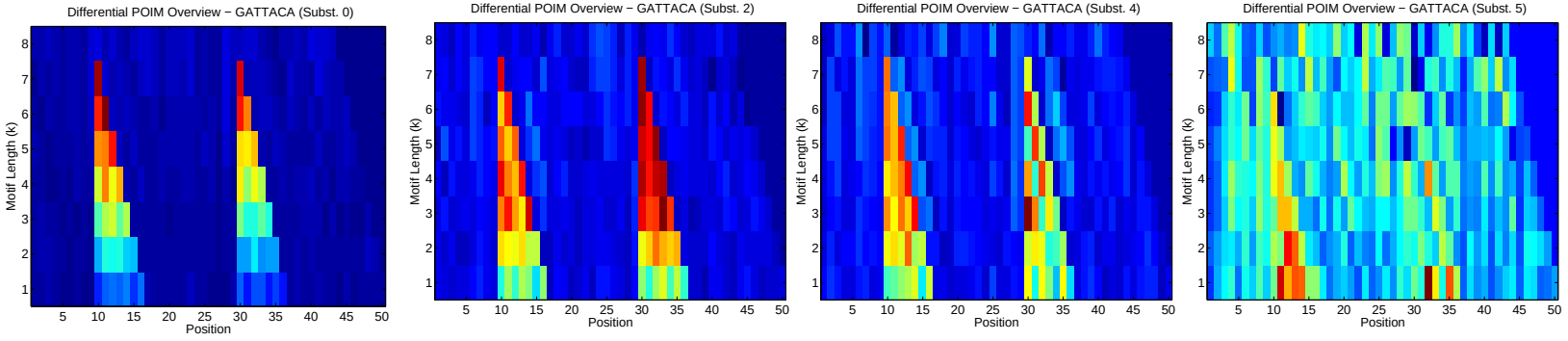
GATTACA and AGTAGTG at fixed positions 10 and 30

TGAGCGCGTGATTACAGTCCGTCTGGGCCAGTAGTGCGTAGTCGCCGGGA
GGCATGGTCGATTACAAACGAGCCCTCTCAGTAGTGGGGGAGCCACGAAA
CCCGTCGAAGATTACACACGGGGCGTGGGAGTAGTGGCGATTACGGGCTC
GGTCGGCAGGATTACACGACGCGTTTACGAGTAGTGAACACTGACTCCTC

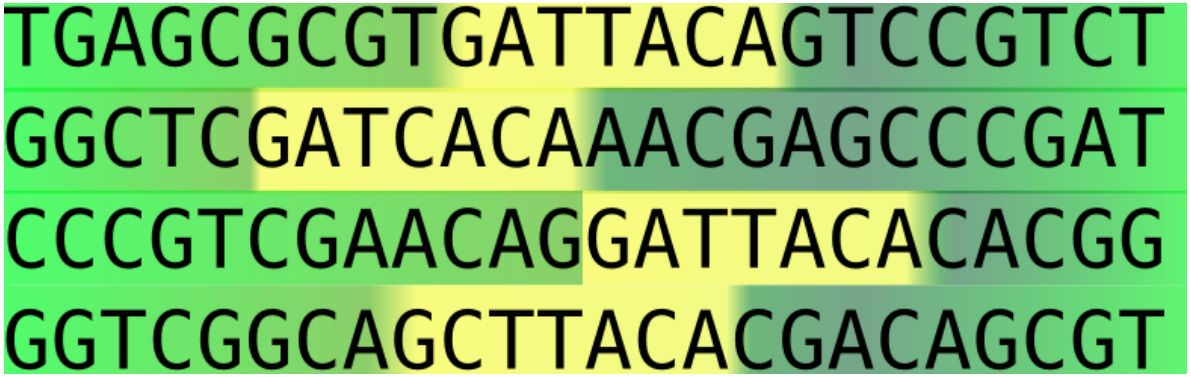
w



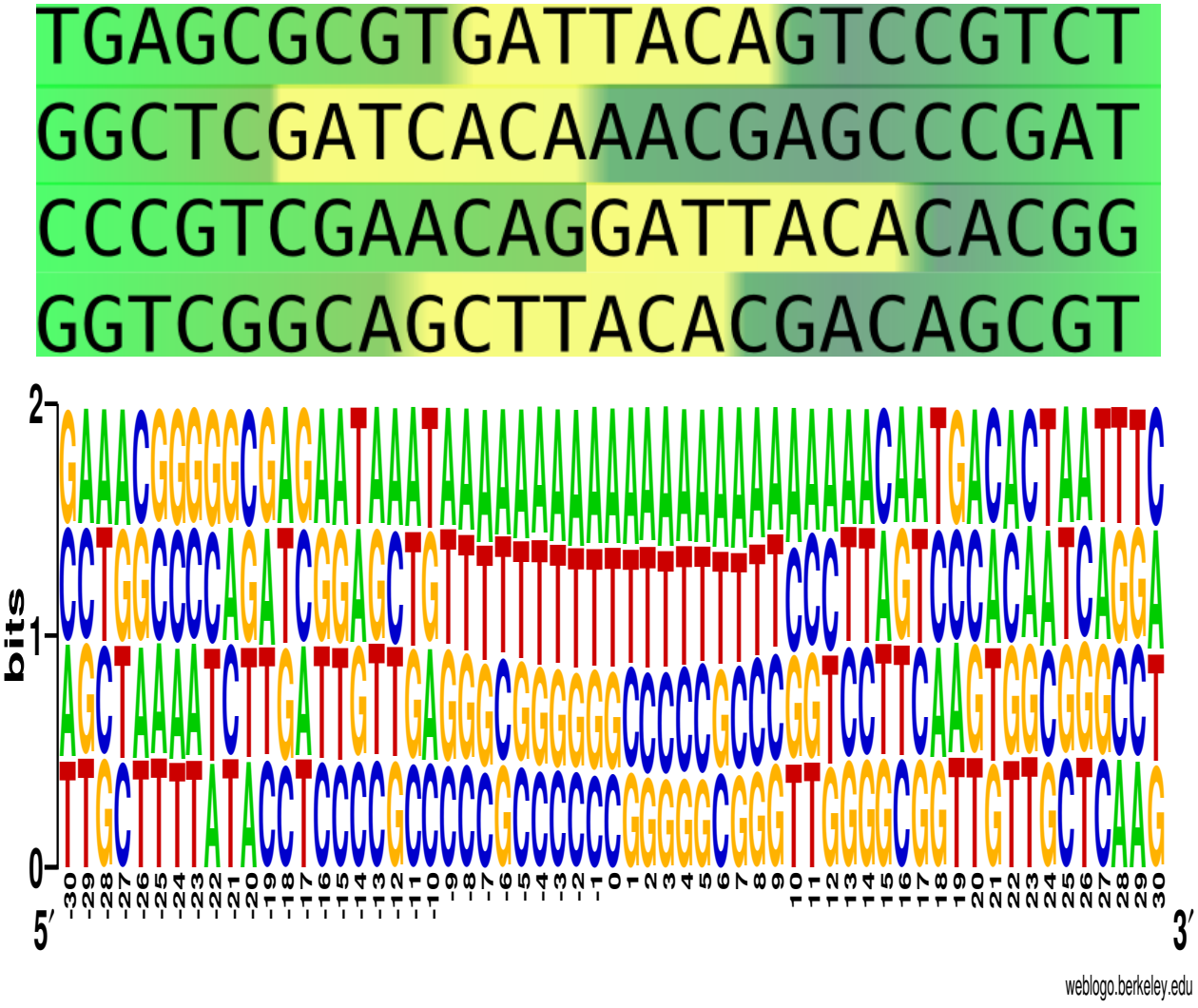
Q



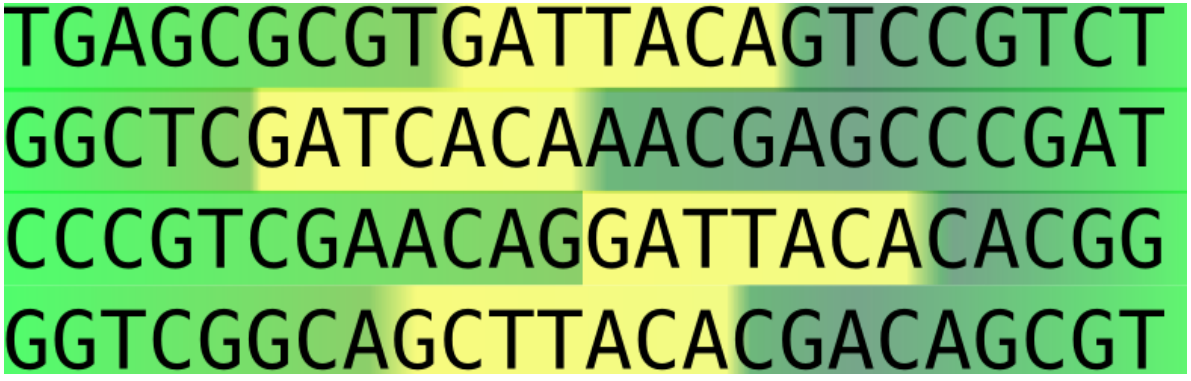
GATTACA at variable positions

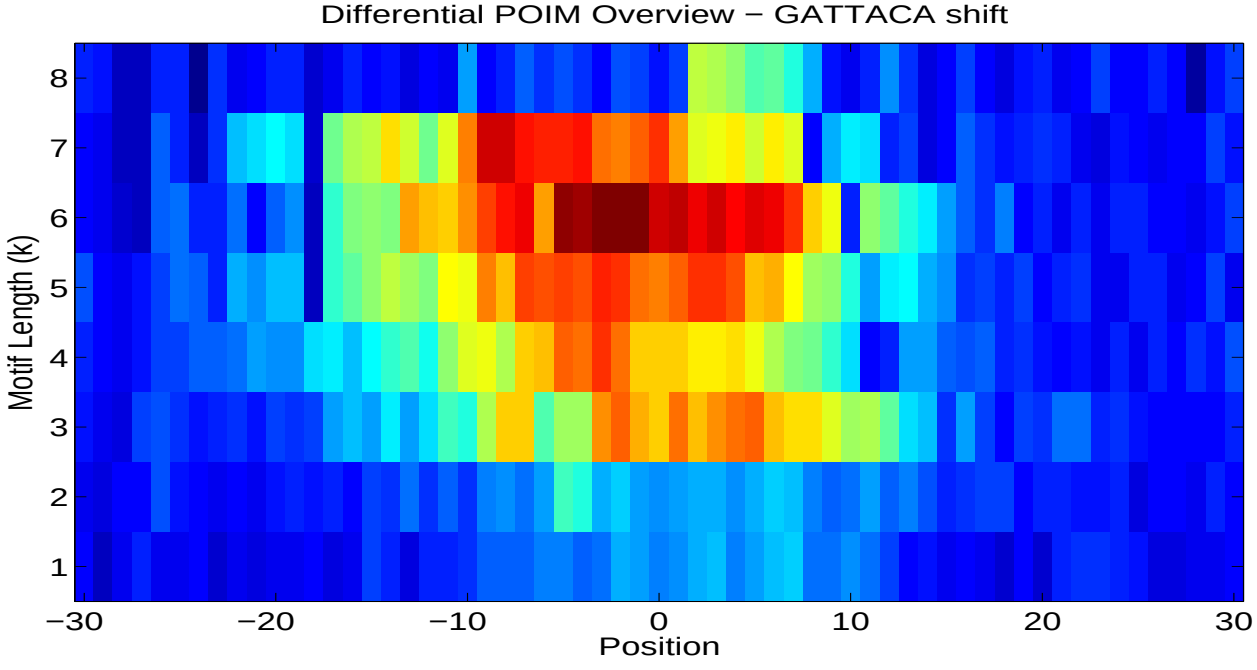


GATTACA at variable positions

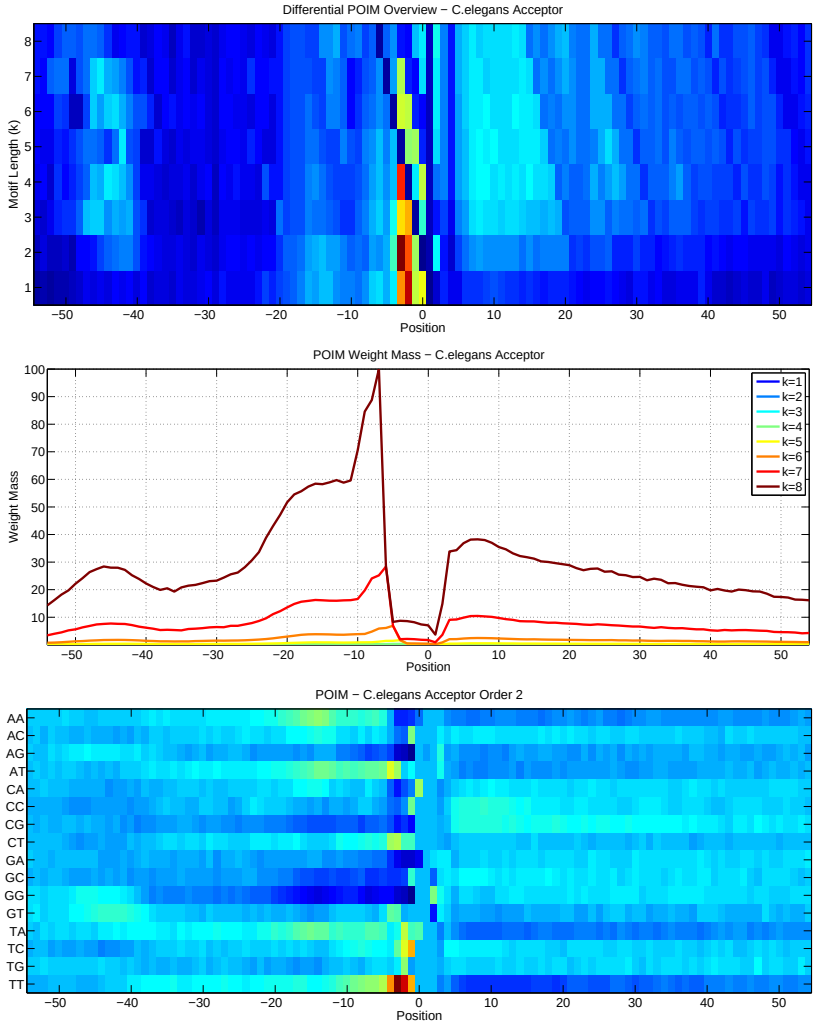


GATTACA at variable positions



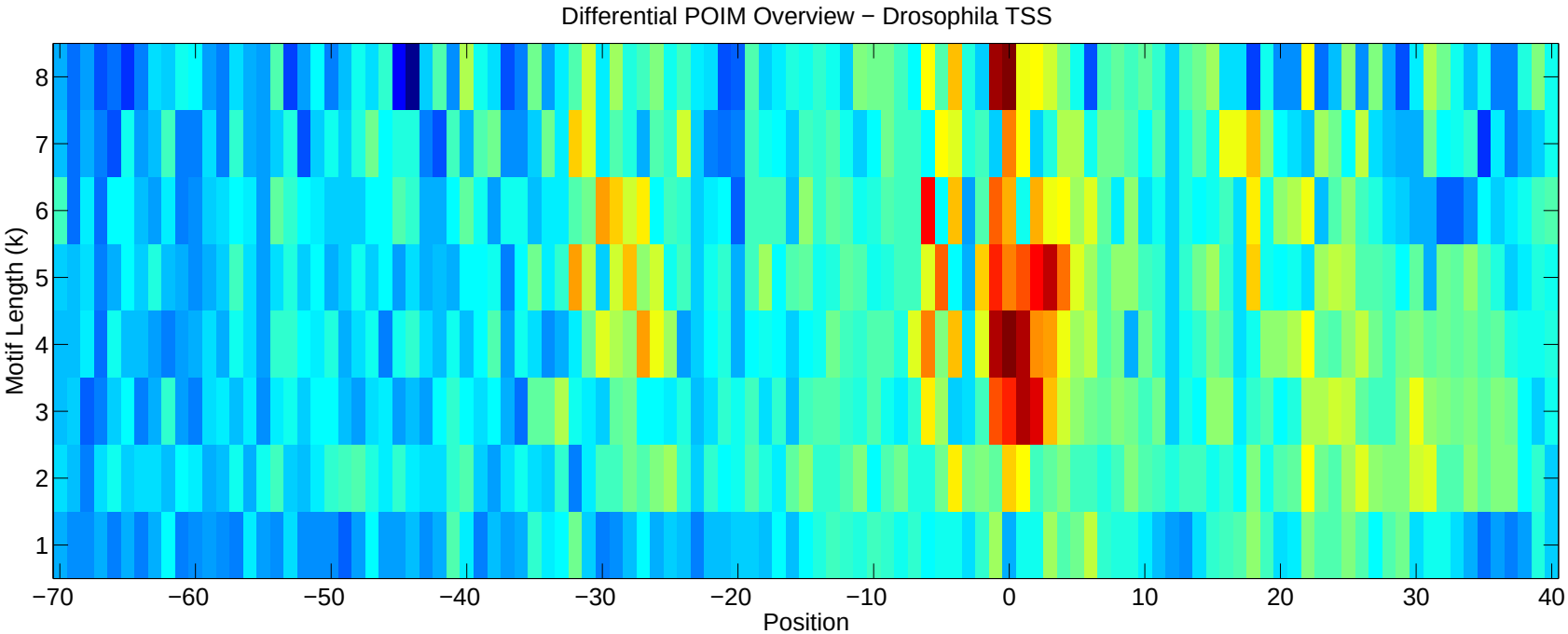


C.elegans Acceptor Splice Site Recognition



- Upstream
  - AG**GT**AAGT -44/++ Donor
  - GGGGGG -16/- - Silencer?
  - **TAAT**AA -16/++ Branch
- Central
  - TTTTTTC -06/+
  - TTTC**AG**$\frac{A}{G}$ -03/++ Acceptor
- Downstream
  - TTTTTTTT +07/- -
  - TTTTT +26/- -

Drosophila Transcription Starts



TATAAAA	-29/++	**CAGT**CAGT	-01/++	CGTCGCG	+18/++
GTATAAA	-30/++	T**CAGT**TGT	-01/++	GCGCGCG	+23/++
ATATAAA	-28/++	CGT**CAGT**T	-03/++	CGCGCGC	+22/++
TATA-box		**Inr** **TCA**$\frac{G}{T}$**T**$\frac{T}{C}$		**CpG**	

Outline

Conclusions

Support Vector Machines with string kernels

- General
- Often are state-of-the art signal detectors
- Applicable to genome-sized datasets
- Using POIMs SVMs are interpretable
  - Importances of positional motifs for the expected decision score
  - Useful to rank motifs and for visualization
  - Can even identify motif length
  - Applicable for a large class of popular scores (SVM+Spectrum/WD/Oligo kernel; Markov Chain)

Efficient implementation `http://www.shogun-toolbox.org`
More machine learning software `http://mloss.org`