Genomic Signal Detection

. . . *using Support Vector Machines*

Sören Sonnenburg
TU Berlin

joint work with
Alexander Zien, Jonas Behr, Gabriele Schweikert,
Konrad Rieck, Petra Philips, Gunnar Rätsch, Vojtech Franc

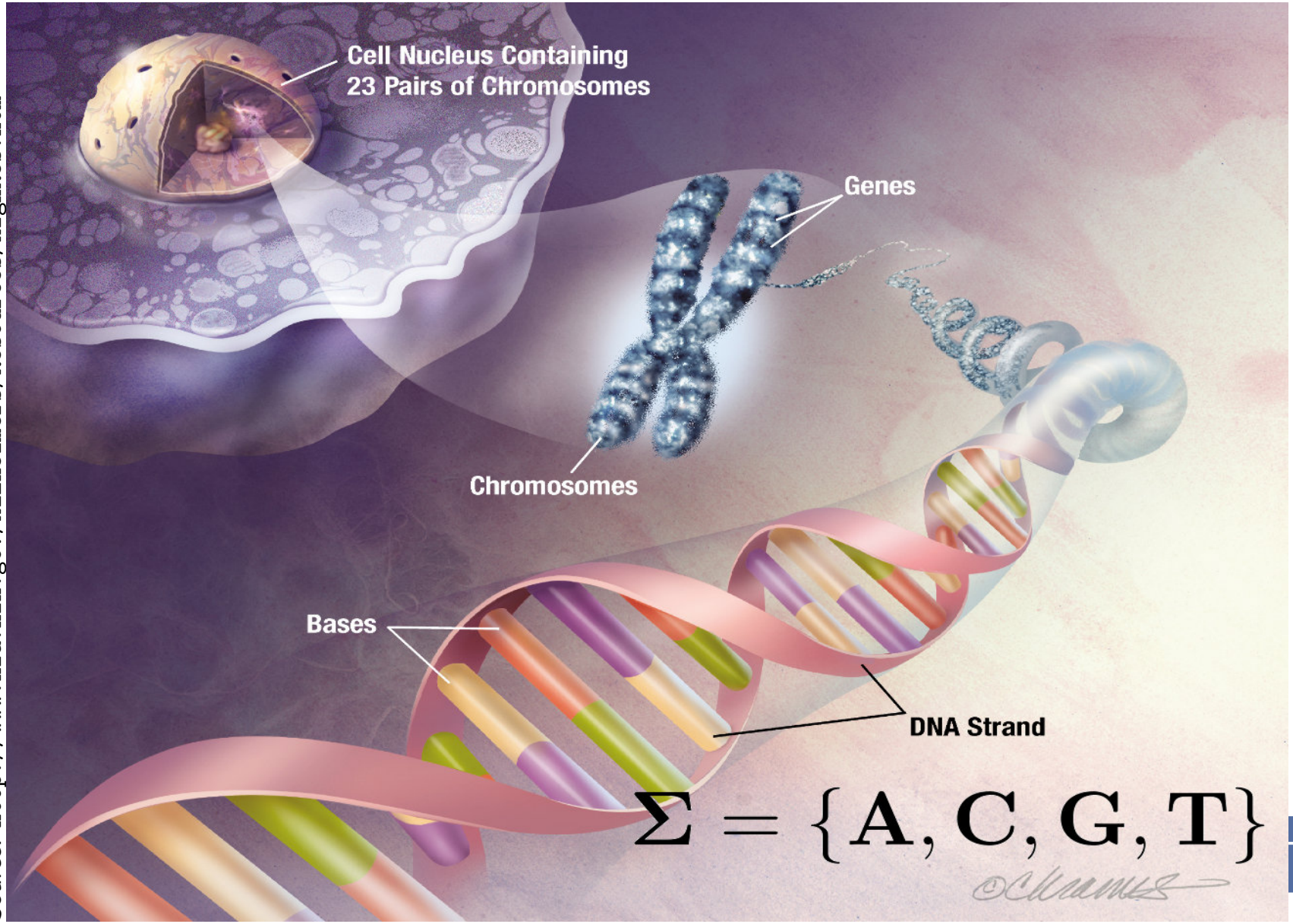
Outline

1. Introduction
2. Sequence Classification
3. Support Vector Machines
4. Applications
5. Explanation and Visualization
6. Summary

Outline

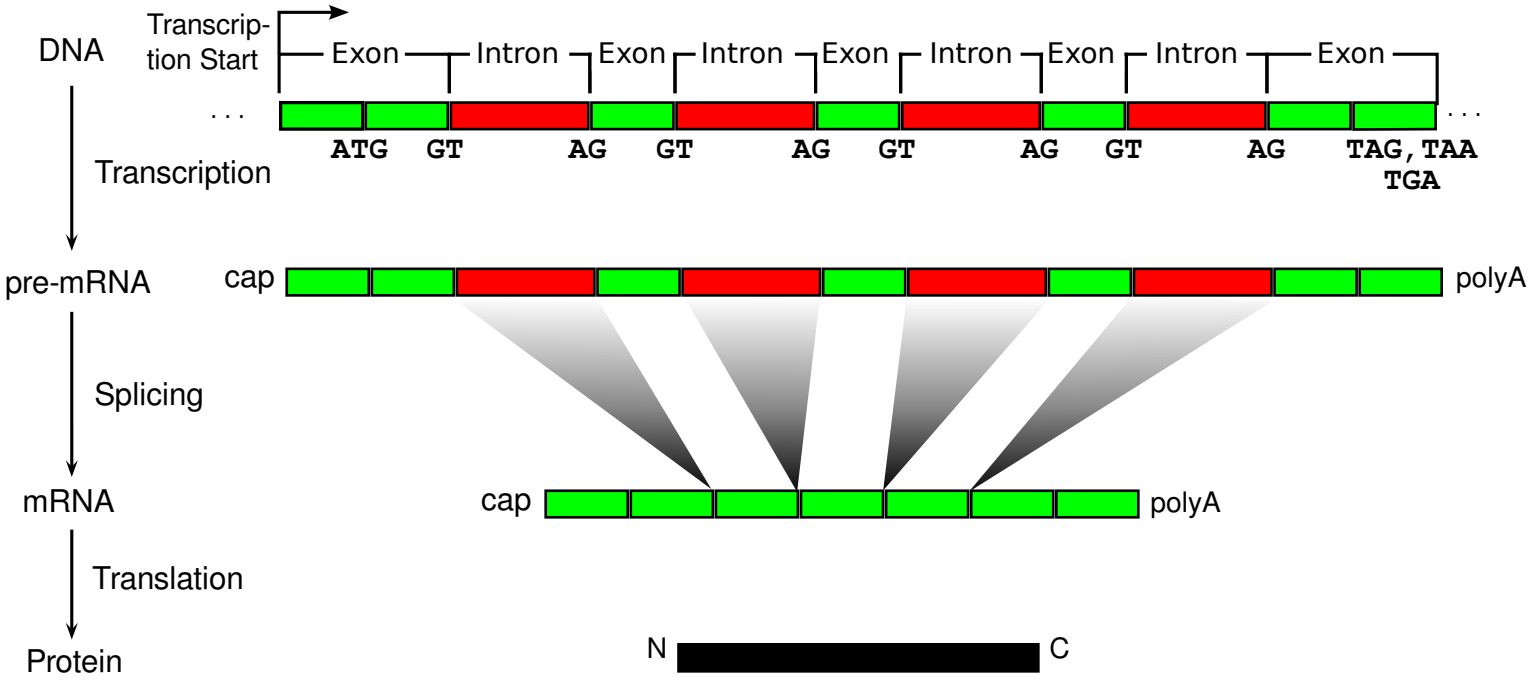
1. Introduction
   - Genomic Signals
2. Sequence Classification
   - Support Vector Machines
3. Support Vector Machines
4. Applications
   - Splice Site Recognition
   - TSS Recognition
   - Gene Finding - mGene
   - Aligning Short Reads - QPALMA
5. Explanation and Visualization
   - Introduction
   - Definition
   - Applications
6. Summary

Genome



Source: http://www.nia.nih.gov/Alzheimers/Resources/HighRes.htm

Genomic Signals

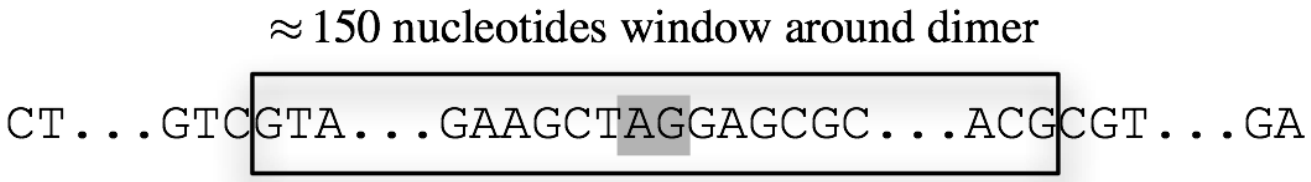


Genomic Signal Detection

- Start/Stop of Genes
- Donor Splice Site (Exon-Intron-Boundary)
- Acceptor Splice Site (Intron-Exon-Boundary)

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

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

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

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,

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

Outline

1. Introduction
   - Genomic Signals
2. Sequence Classification
   - Support Vector Machines
3. Support Vector Machines
4. Applications
   - Splice Site Recognition
   - TSS Recognition
   - Gene Finding - mGene
   - Aligning Short Reads - QPALMA
5. Explanation and Visualization
   - Introduction
   - Definition
   - Applications
6. Summary

Classification - Learning based on examples I

Given:

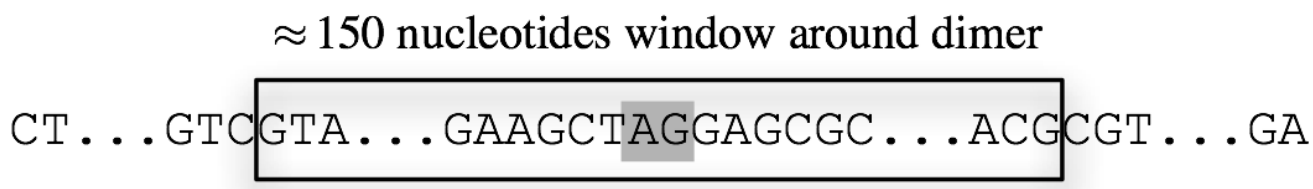
Training examples $(\mathbf{x}_i, y_i)_{i=1}^N \in (\{A, C, G, T\}^L, \{-1, +1\})^N$

AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG	$+1$
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT	-1
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC	$+1$
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC	-1
TTGTTTTAATATTCAATTTTTTACAGTAAGTTGCCAATTCAATGTTCCAC	-1
TACCTAATTATGAAATTAAAATTCAGTGTGCTGATGGAAACGGAGAAGTC	-1

($\approx$ 1 billion neg. sequences; $<$ 200.000 positive sequences)

Wanted:

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



Aim: Accurate signal prediction for the whole genome

Classification - Learning based on examples I

Given:

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

AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG	$+1$
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT	-1
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC	$+1$
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC	-1
TTGTTTTAATATTCAATTTTTTACAGTAAGTTGCCAATTCAATGTTCCAC	-1
TACCTAATTATGAAATTAAAATTCAGTGTGCTGATGGAAACGGAGAAGTC	-1

($\approx$ 1 billion neg. sequences; $<$ 200.000 positive sequences)

Wanted:

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

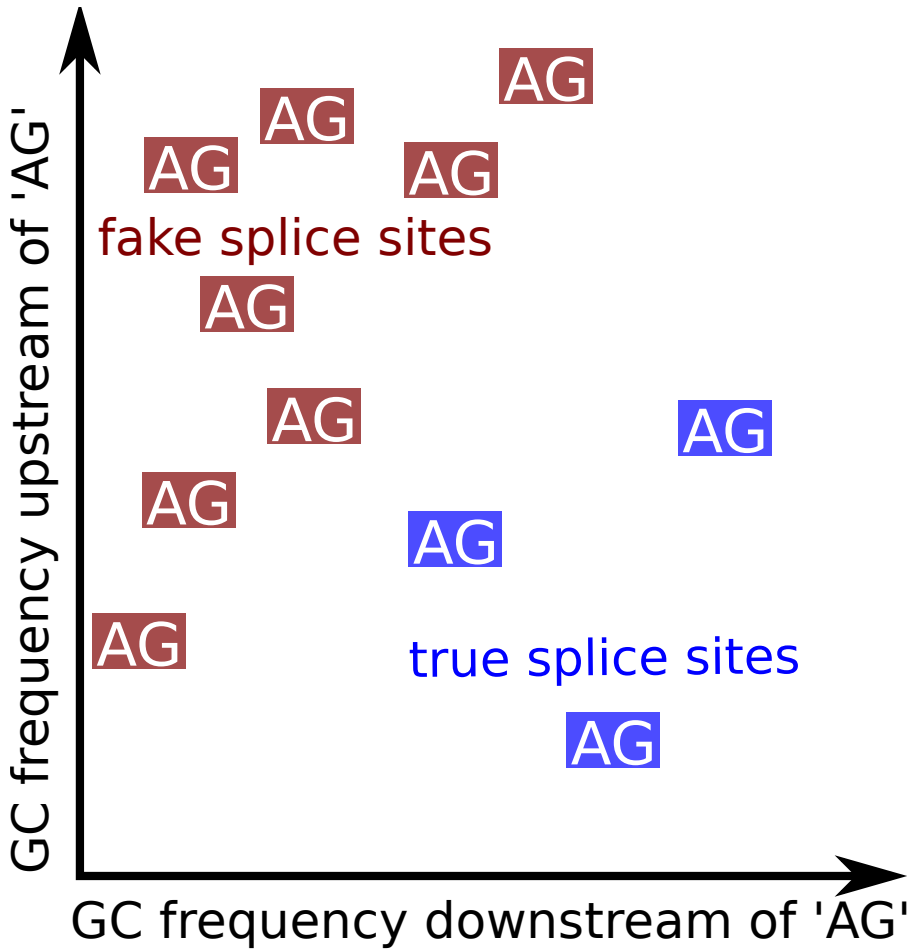
$\approx$ 150 nucleotides window around dimer

CT...GTCGTA...GAAGCTAGGAGCGC...ACGCGT...GA

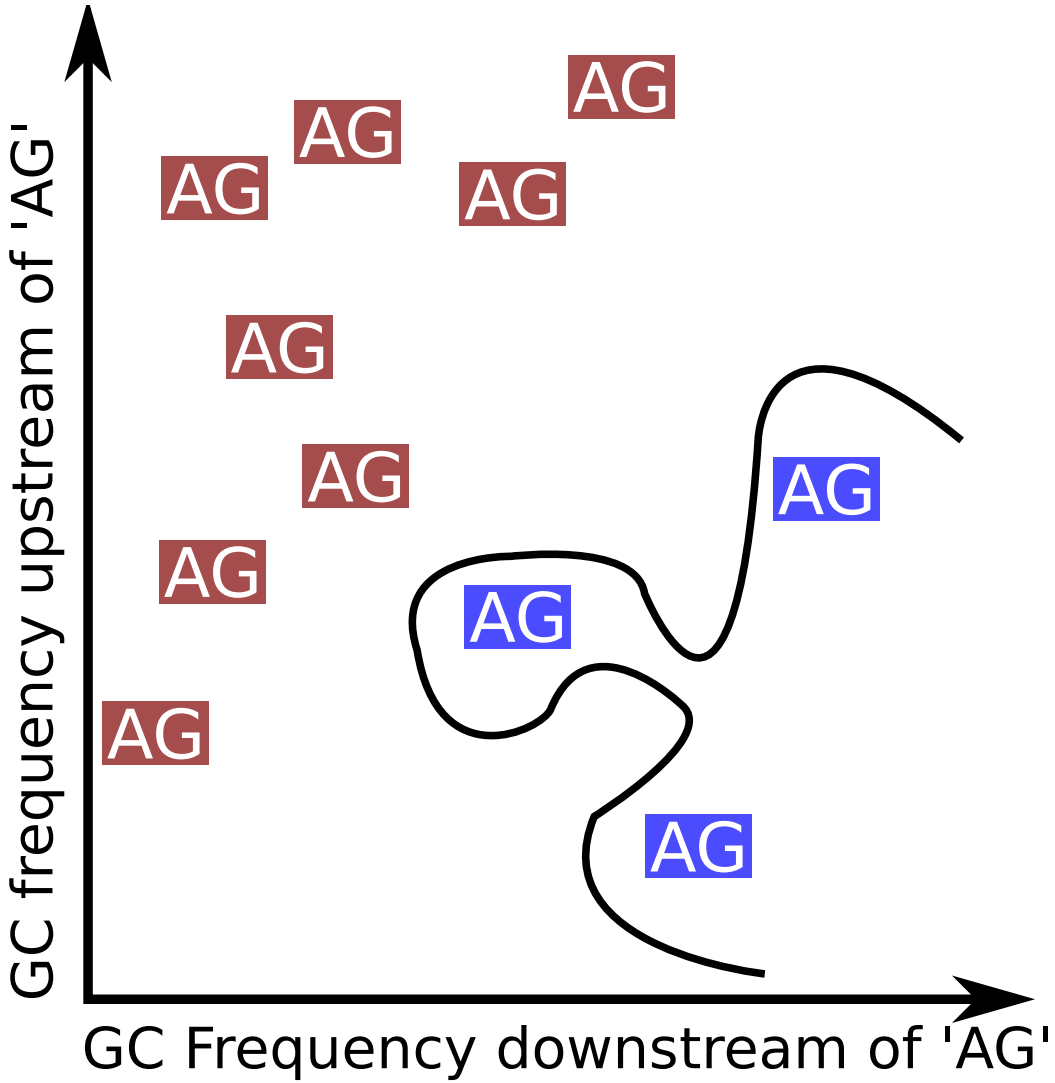
Aim: Accurate signal prediction for the whole genome

Classification - Learning based on examples II

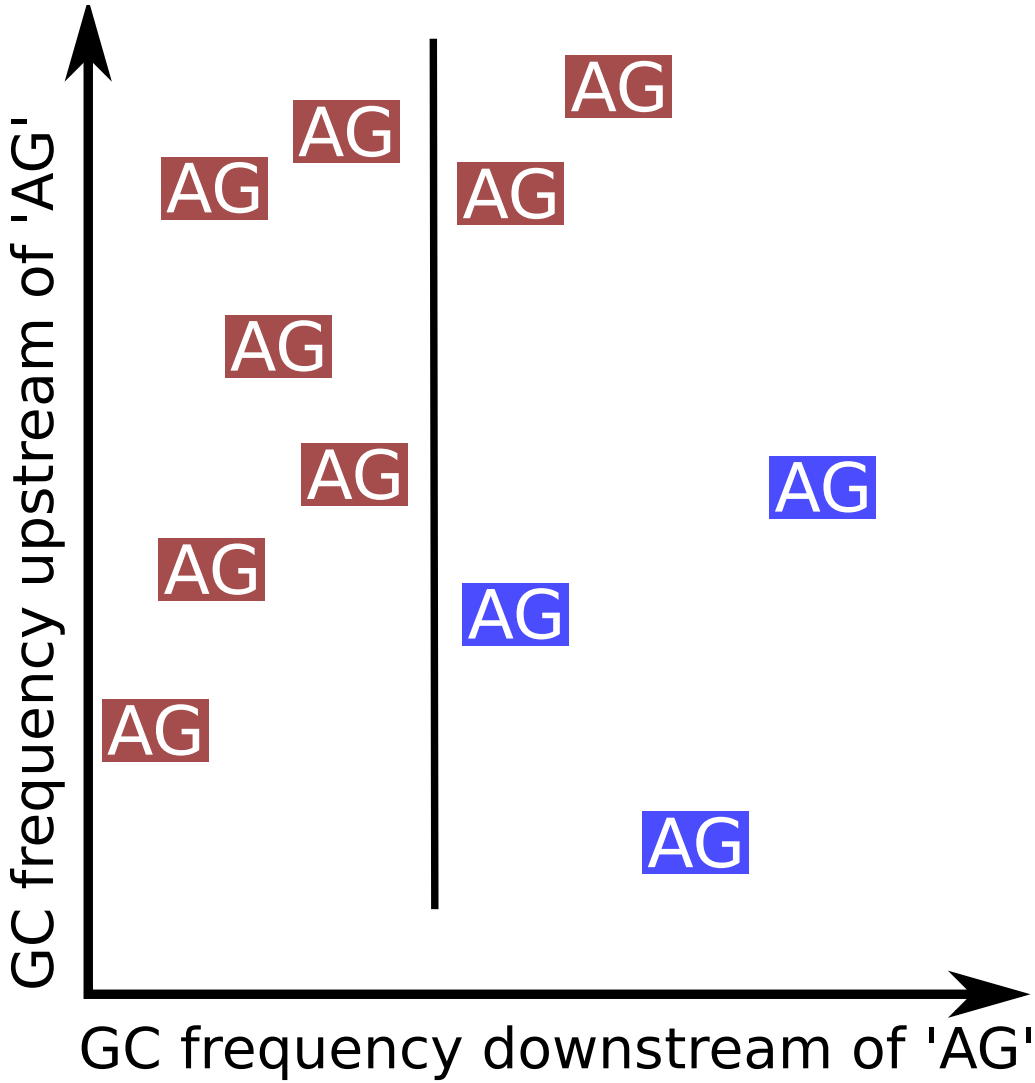
AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG



Classification - Learning based on examples III



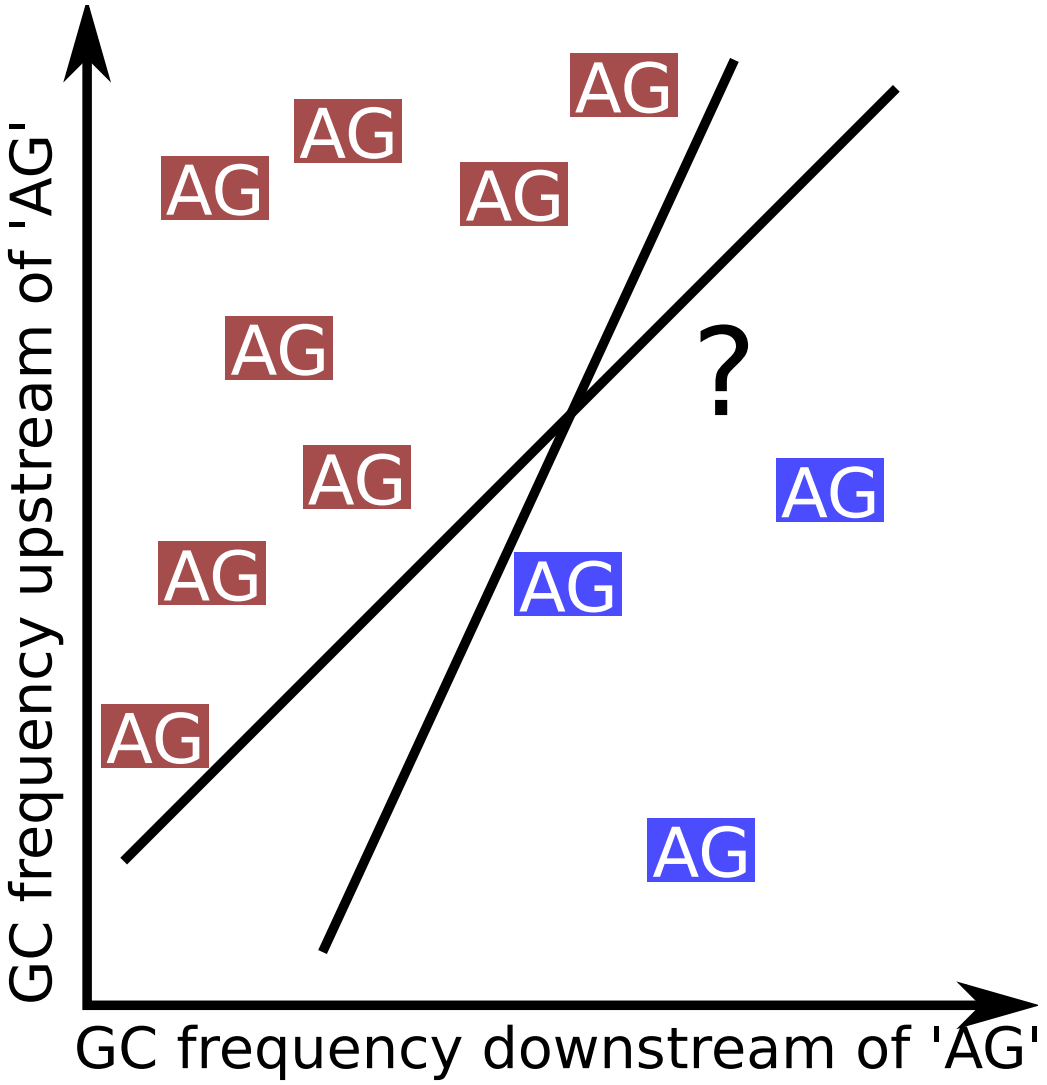
Classification - Learning based on examples IV



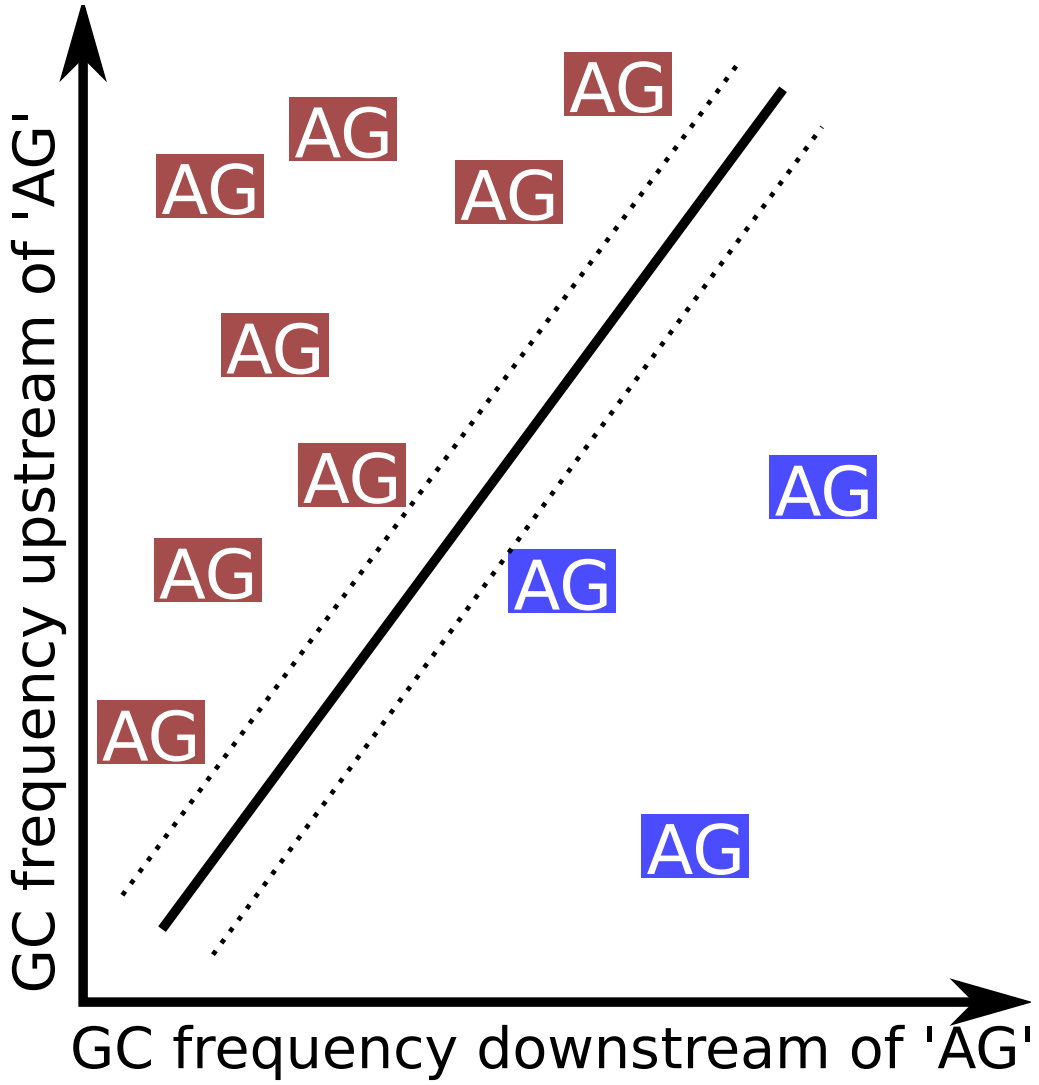
Outline

1. Introduction
   - Genomic Signals
2. Sequence Classification
   - Support Vector Machines
3. Support Vector Machines
4. Applications
   - Splice Site Recognition
   - TSS Recognition
   - Gene Finding - mGene
   - Aligning Short Reads - QPALMA
5. Explanation and Visualization
   - Introduction
   - Definition
   - Applications
6. Summary

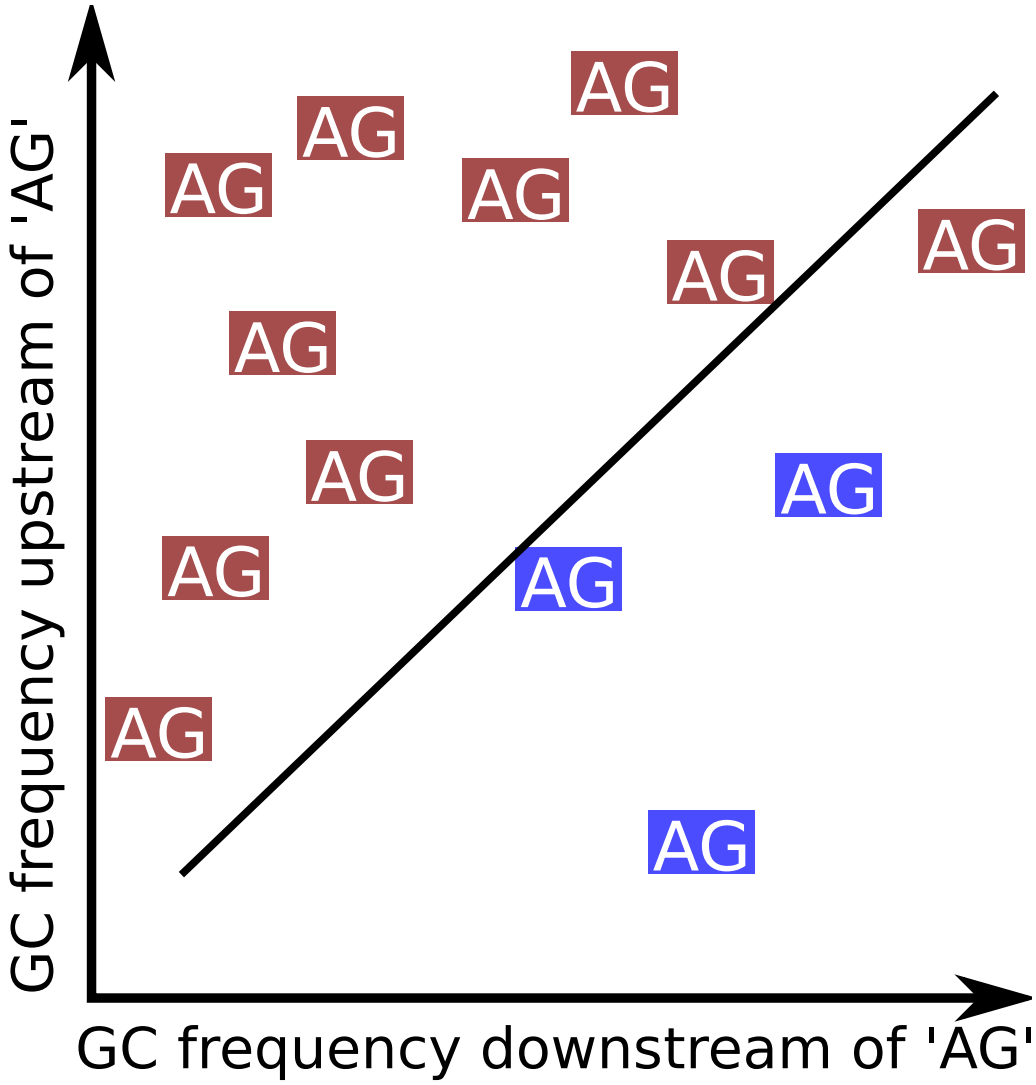
Support Vector Machines (SVMs) I



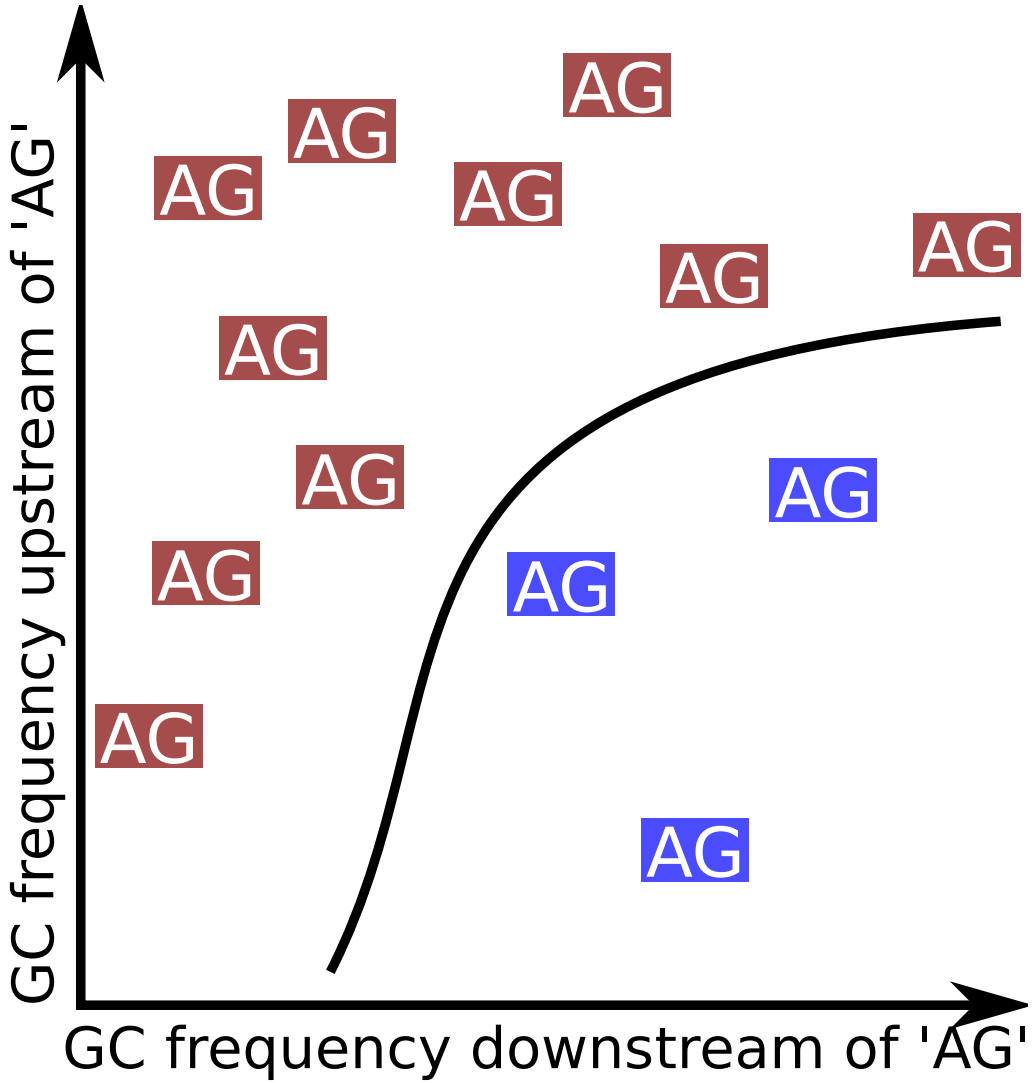
Support Vector Machines (SVMs) II



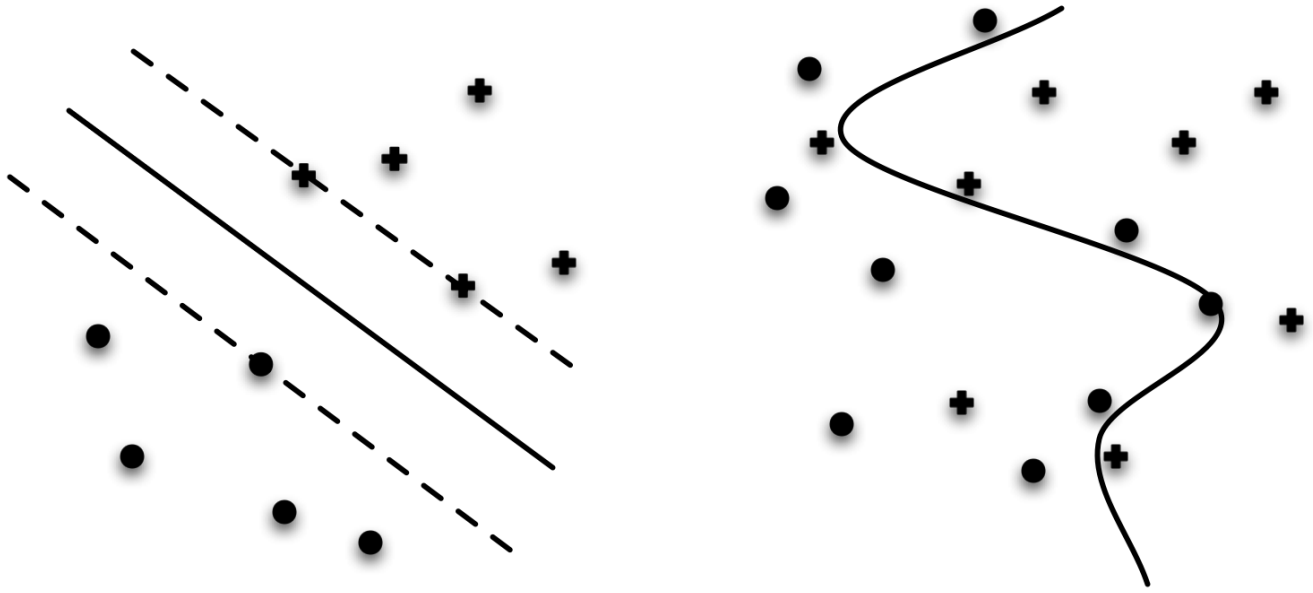
Support Vector Machines (SVMs) III



SVMs and Kernels



Support Vector Machines (SVMs)



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

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

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

The Spectrum Kernel (Leslie et al. 2002)

Support Vector Machine

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

Spectrum Kernel (with mismatches, gaps)

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

Example $k = 3$:

x AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG

x' TACCTAATTATGAAATTAAATTTCAGTGTGCTGATGGAAACGGAGAAGTC

3-mer	AAA	AAC	...	CCA	CCC	...	TTT
# in $\mathbf{x}$	2	4	...	1	0	...	3
# in $\mathbf{x}'$	3	1	...	0	0	...	1

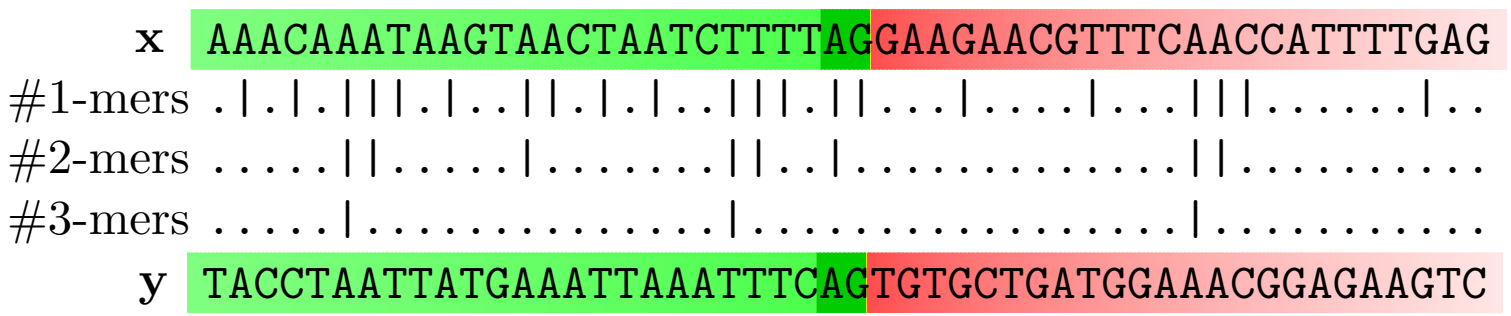
$$\mathrm{k}(\mathbf{x}, \mathbf{x}') = 2 \cdot 3 + 4 \cdot 1 + \ldots 1 \cdot 0 + 0 \cdot 0 \ldots 3 \cdot 1$$

The Weighted Degree Kernel (Sonnenburg et al. 2005)

Support Vector Machine

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

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

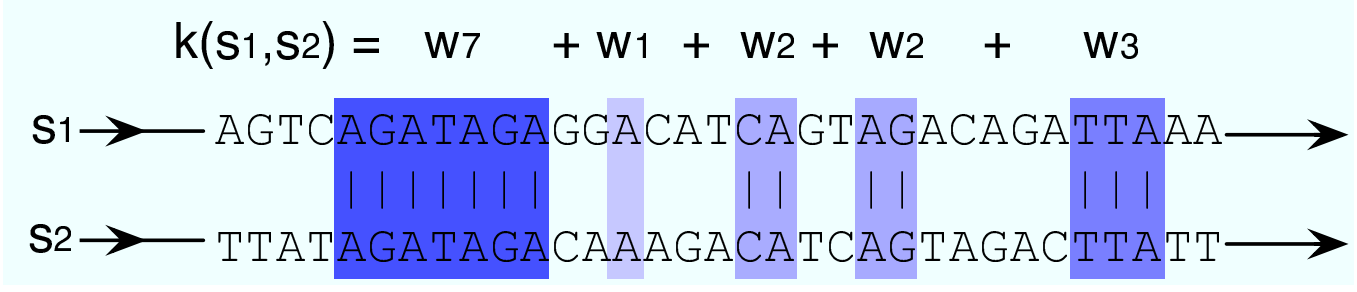


Example: $K = 3$: $\mathsf{k}(\mathbf{x}, \mathbf{x}') = \beta_1 \cdot 21 + \beta_2 \cdot 8 + \beta_3 \cdot 3$

The Weighted Degree Kernel with *shifts* (Raetsch, Sonnenburg et al. 2005)

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

Aim: Train and apply string-kernel SVM on all available data

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

Idea of the `Linadd` Algorithm (Sonnenburg et al., 2005):

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

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

Possible for low-dimensional or sparse

Effort: $\mathcal{O}(ML) \Rightarrow$ **speedup of factor** N_s (with $L := dim(\mathcal{X})$)

General Recipe — Howto

Collect Data

AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG	$+1$
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT	-1
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC	$+1$
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC	-1
TTGTTTTAATATTCAATTTTTTACAGTAAGTTGCCAATTCAATGTTCCAC	-1
TACCTAATTATGAAATTAAAATTCAGTGTGCTGATGGAAACGGAGAAGTC	-1

What features can be used to describe "my Signal"?

- Are these *position independent* k-mers? $\Rightarrow$ Spectrum Kernel?
- Are these *strongly position dependent* k-mers? $\Rightarrow$ WD Kernel?
- Are these *partially position dependent* k-mers? $\Rightarrow$ WDK *with shifts*

Train SVM

- split data into training, validation, test
- model selection over hyperparameters C, k, s
- learn final model on train+validation, estimate performance on test

General Recipe — Howto

Collect Data

AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG	$+1$
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT	-1
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC	$+1$
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC	-1
TTGTTTTAATATTCAATTTTTTACAGTAAGTTGCCAATTCAATGTTCCAC	-1
TACCTAATTATGAAATTAAAATTCAGTGTGCTGATGGAAACGGAGAAGTC	-1

What features can be used to describe "my Signal"?

- Are these *position independent* k-mers? $\Rightarrow$ Spectrum Kernel?
- Are these *strongly position dependent* k-mers? $\Rightarrow$ WD Kernel?
- Are these *partially position dependent* k-mers? $\Rightarrow$ WDK *with shifts*

Train SVM

- split data into training, validation, test
- model selection over hyperparameters C, k, s
- learn final model on train+validation, estimate performance on test

General Recipe — Howto

Collect Data

AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG	$+1$
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT	-1
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC	$+1$
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC	-1
TTGTTTTAATATTCAATTTTTTACAGTAAGTTGCCAATTCAATGTTCCAC	-1
TACCTAATTATGAAATTAAAATTCAGTGTGCTGATGGAAACGGAGAAGTC	-1

What features can be used to describe "my Signal"?

- Are these *position independent* k-mers? $\Rightarrow$ Spectrum Kernel?
- Are these *strongly position dependent* k-mers? $\Rightarrow$ WD Kernel?
- Are these *partially position dependent* k-mers? $\Rightarrow$ WDK *with shifts*

Train SVM

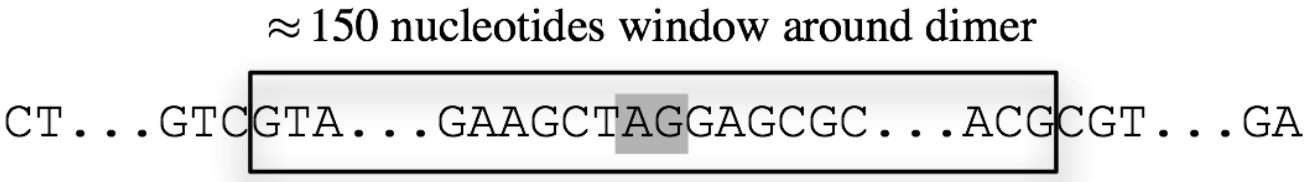
- split data into training, validation, test
- model selection over hyperparameters C, k, s
- learn final model on train+validation, estimate performance on test

Outline

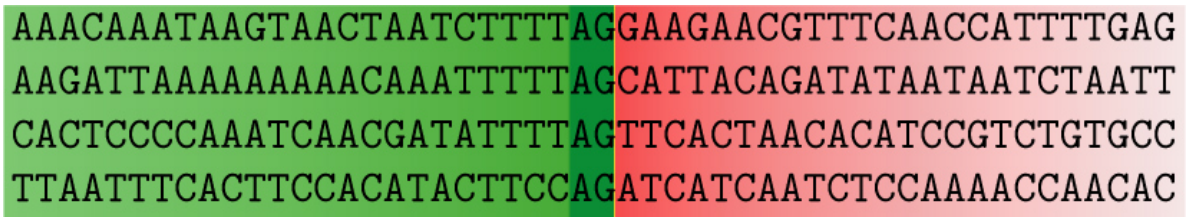
1. Introduction
   - Genomic Signals
2. Sequence Classification
   - Support Vector Machines
3. Support Vector Machines
4. Applications
   - Splice Site Recognition
   - TSS Recognition
   - Gene Finding - mGene
   - Aligning Short Reads - QPALMA
5. Explanation and Visualization
   - Introduction
   - Definition
   - Applications
6. Summary

Splice Site Recognition

Discriminate true signal positions against all other positions



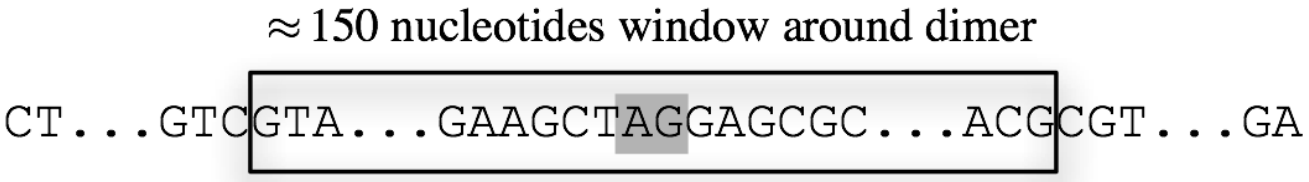
- True sites: fixed window around a true splice site
- Decoy sites: all other consensus sites



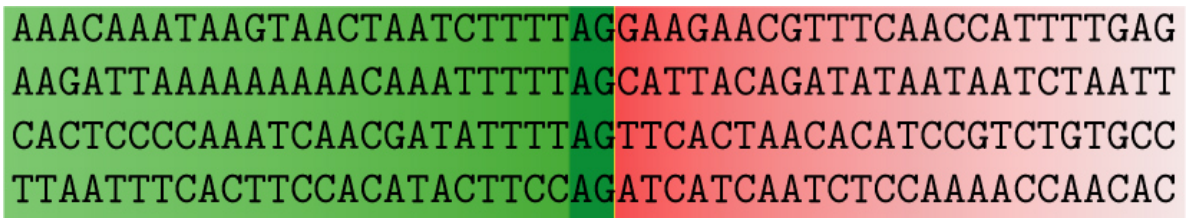
- Create training sample from cDNA/EST alignments to genome (for human, e.g., 50 million examples)
- Sequences are compared via **Weighted Degree Kernel**
- Train SVM on up to 8 million examples.

Splice Site Recognition

Discriminate true signal positions against all other positions



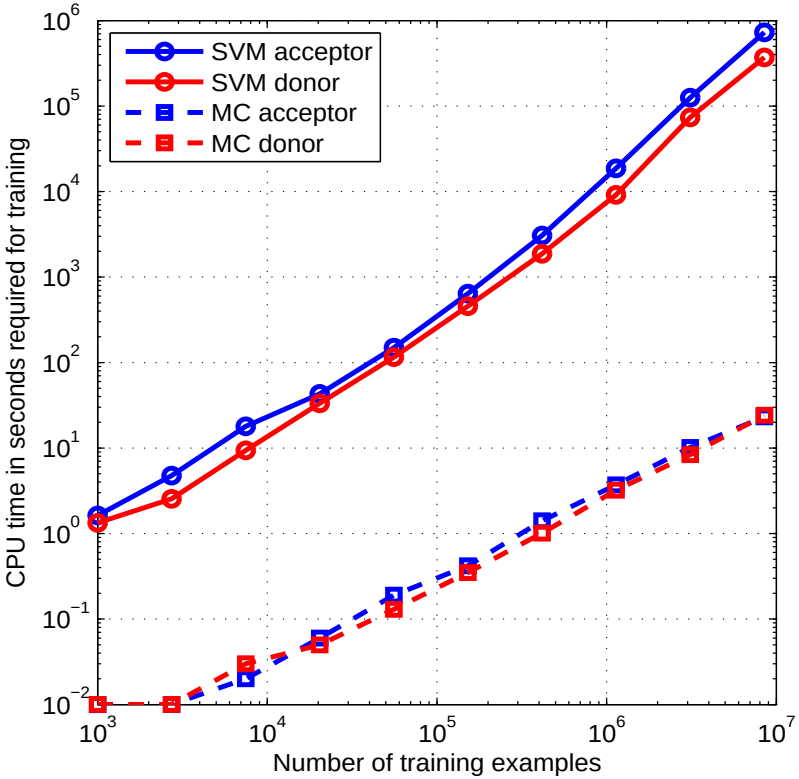
- True sites: fixed window around a true splice site
- Decoy sites: all other consensus sites

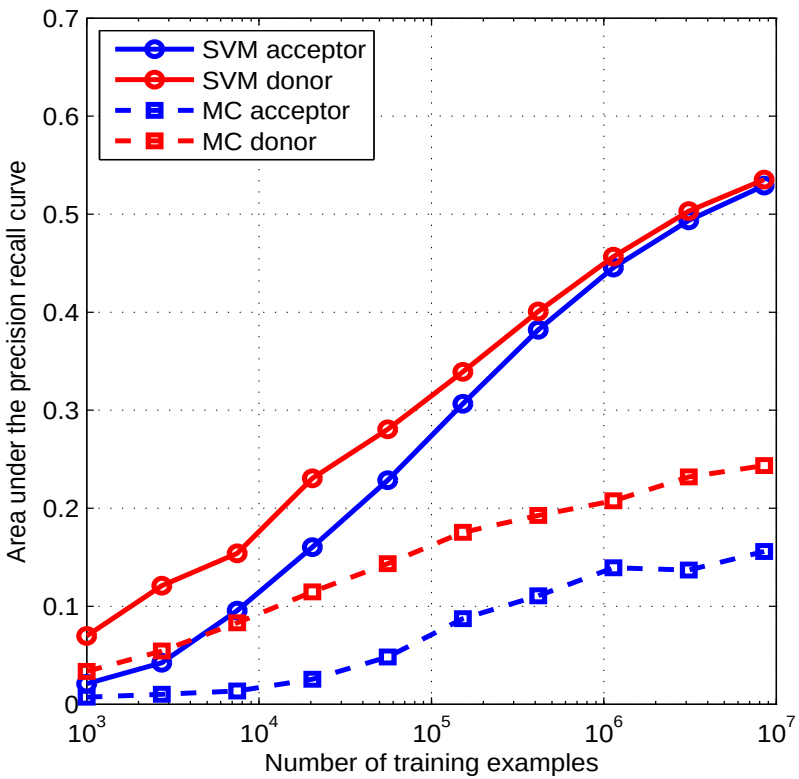


- Create training sample from cDNA/EST alignments to genome (for human, e.g., 50 million examples)
- Sequences are compared via **Weighted Degree Kernel**
- Train SVM on up to 8 million examples.

Splice Site Recognition - Results

- Human splice sites: $5 \cdot 10^7$ strings of length ≈ 141
- Note: Raw data is already 7GB in size

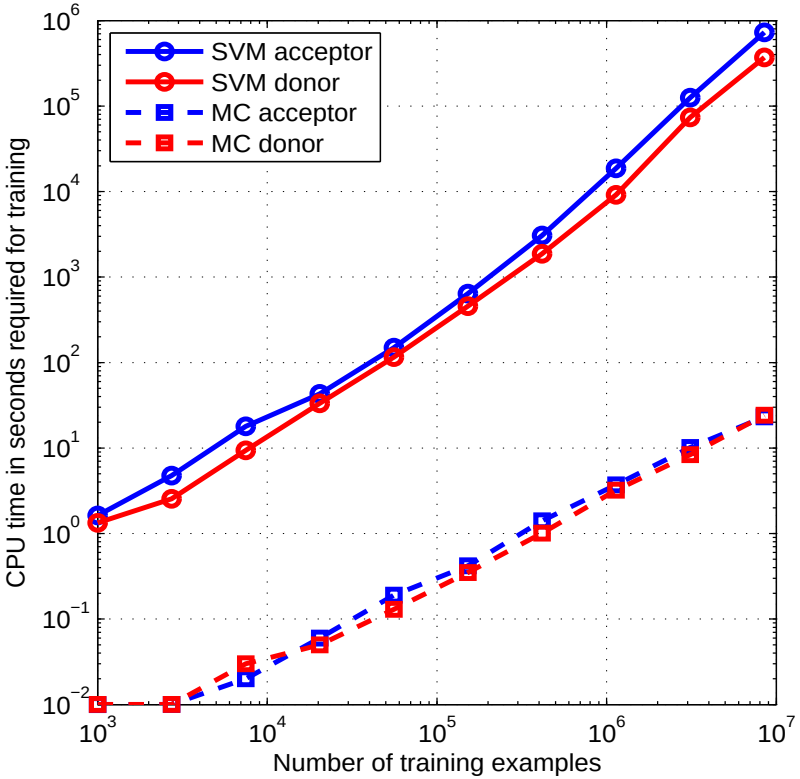


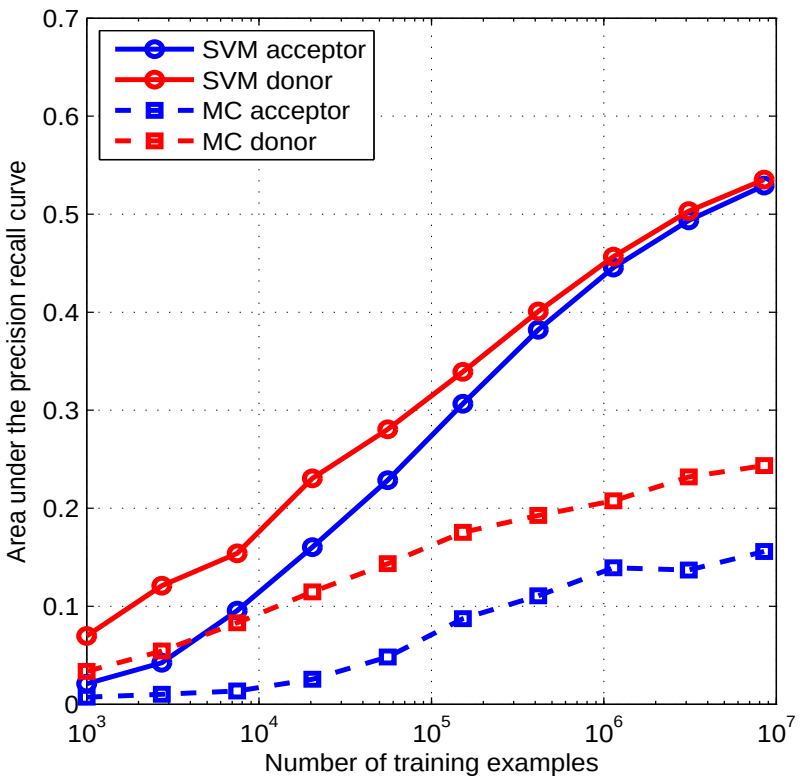


SVM ≈ 3 times more accurate than IMCs
(54.4% vs. 16.2% auPRC; Sonnenburg et al. 2007)

Splice Site Recognition - Results

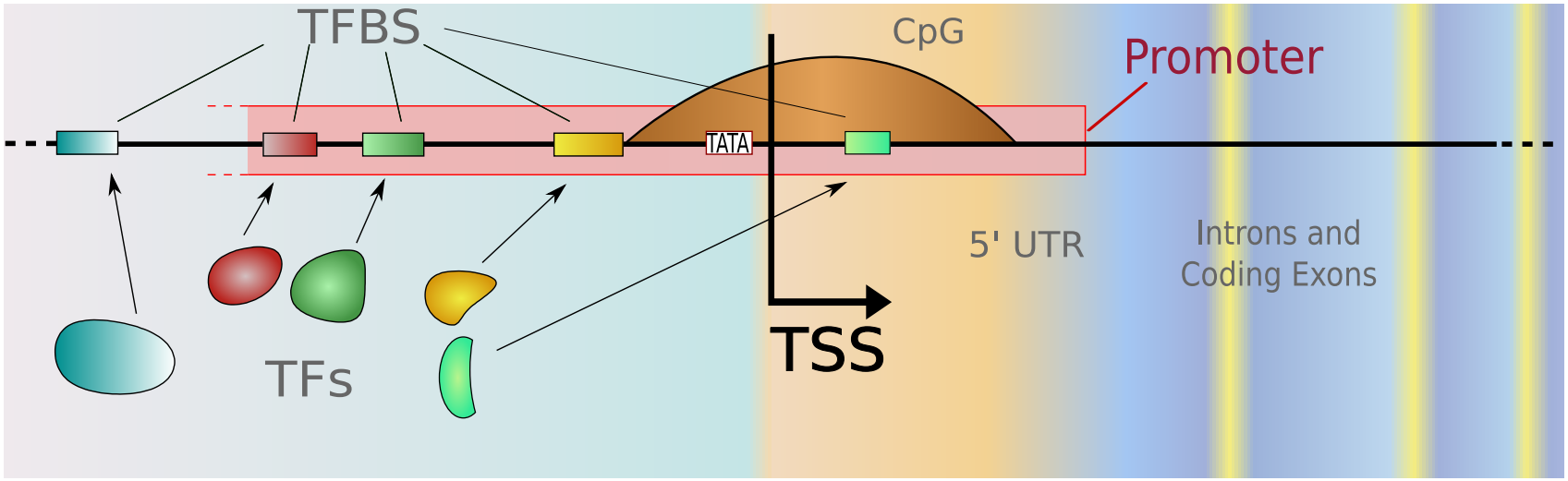
- Human splice sites: $5 \cdot 10^7$ strings of length ≈ 141
- Note: Raw data is already 7GB in size





SVM ≈ 3 times more accurate than IMCs
(54.4% vs. 16.2% auPRC; Sonnenburg et al. 2007)

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)
- TFBS in Promoter region, 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**

Training – Data Generation

Training and Validation Data (50% : 50%)

True TSS (8508 positive)

- From dbTSSv4 (based on hg16) extract putative TSS windows of size $[-1000, +1000]$

Decoy TSS (85042)

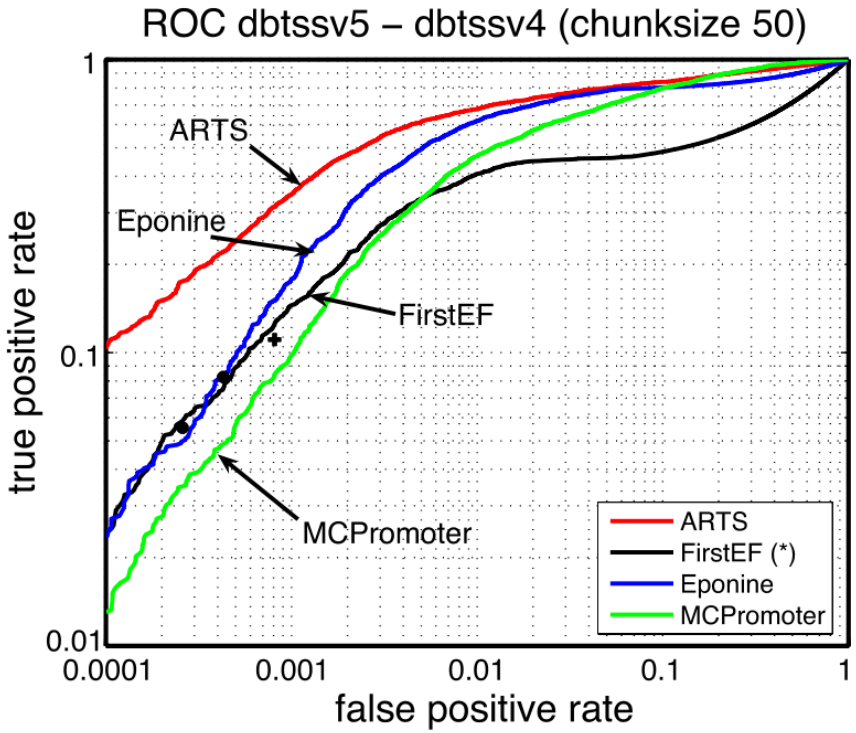
- Annotate dbTSSv4 with transcription-stop (via *BLAT* alignment of mRNAs)
- From the interior of the gene ($+100bp$ to gene end) sample negatives for training (10 per positive)

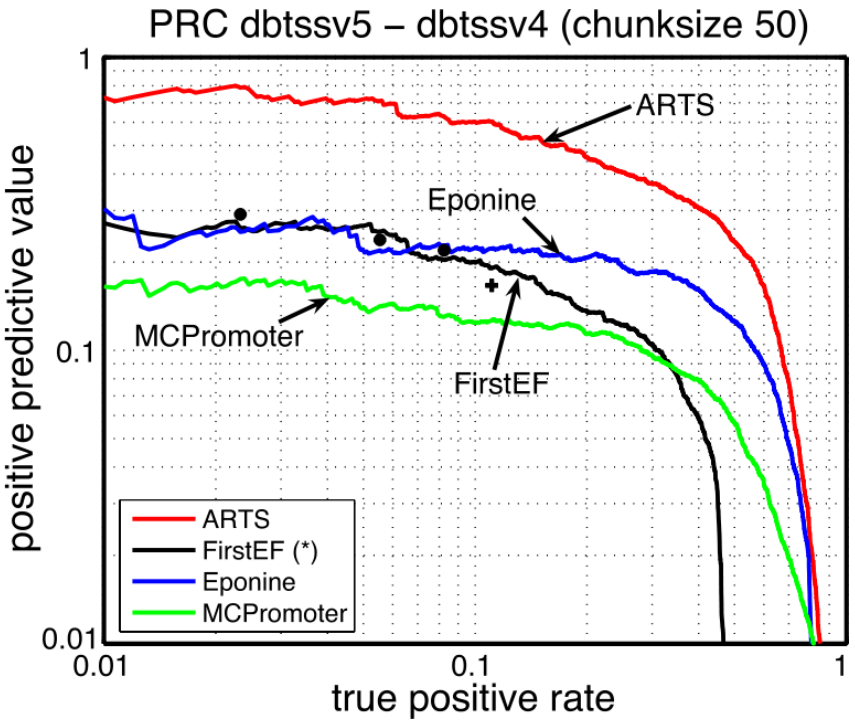
Fair genome-wide evaluation

- Compare against FirstEF, McPromoter, Eponine
- Only consider "new" TSS from dbTSSv5-dbTSSv4, with max 30% overlap

State-of-the-art Performance

Receiver Operator Characteristic and Precision Recall Curve





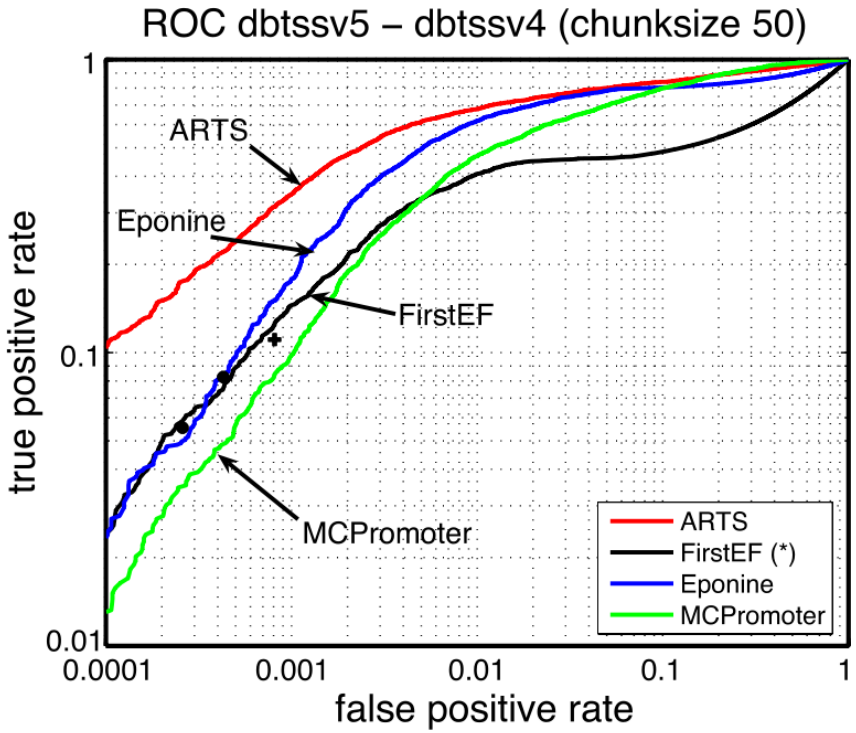
ARTS (Sonnenburg et al. 2006) twice as accurate!

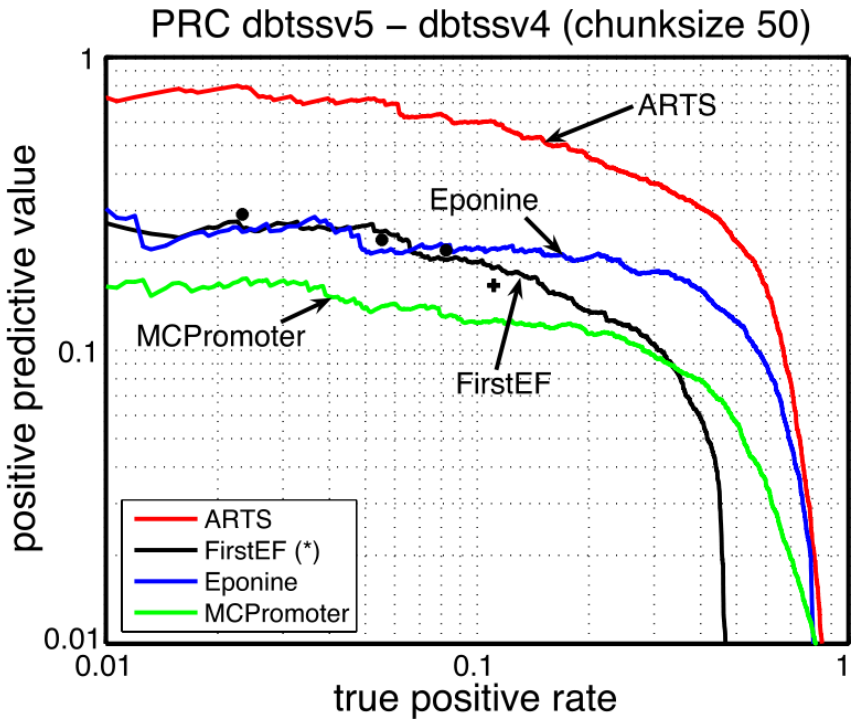
Independent evaluation of 17 methods (Abeel et al. ISMB, 2009)

TSS detector (ARTS) winner in evaluation of 17 methods.

State-of-the-art Performance

Receiver Operator Characteristic and Precision Recall Curve



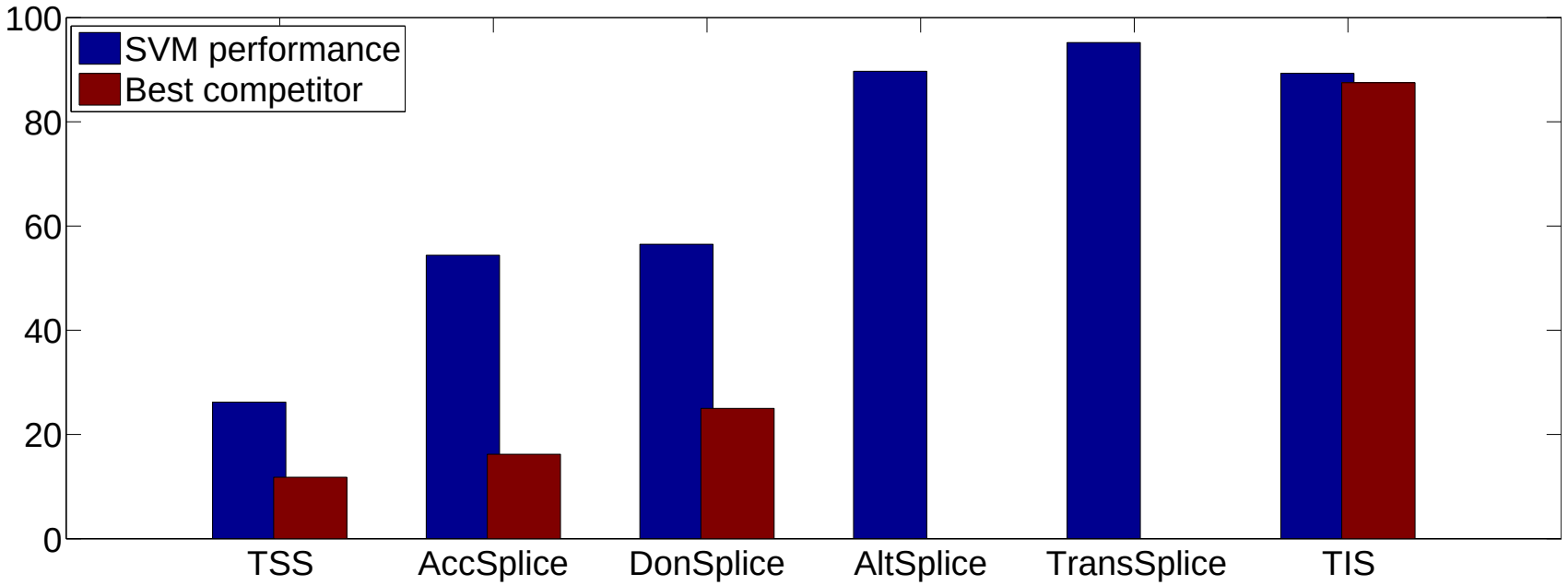


ARTS (Sonnenburg et al. 2006) twice as accurate!

Independent evaluation of 17 methods (Abeel et al. ISMB, 2009)

TSS detector (ARTS) winner in evaluation of 17 methods.

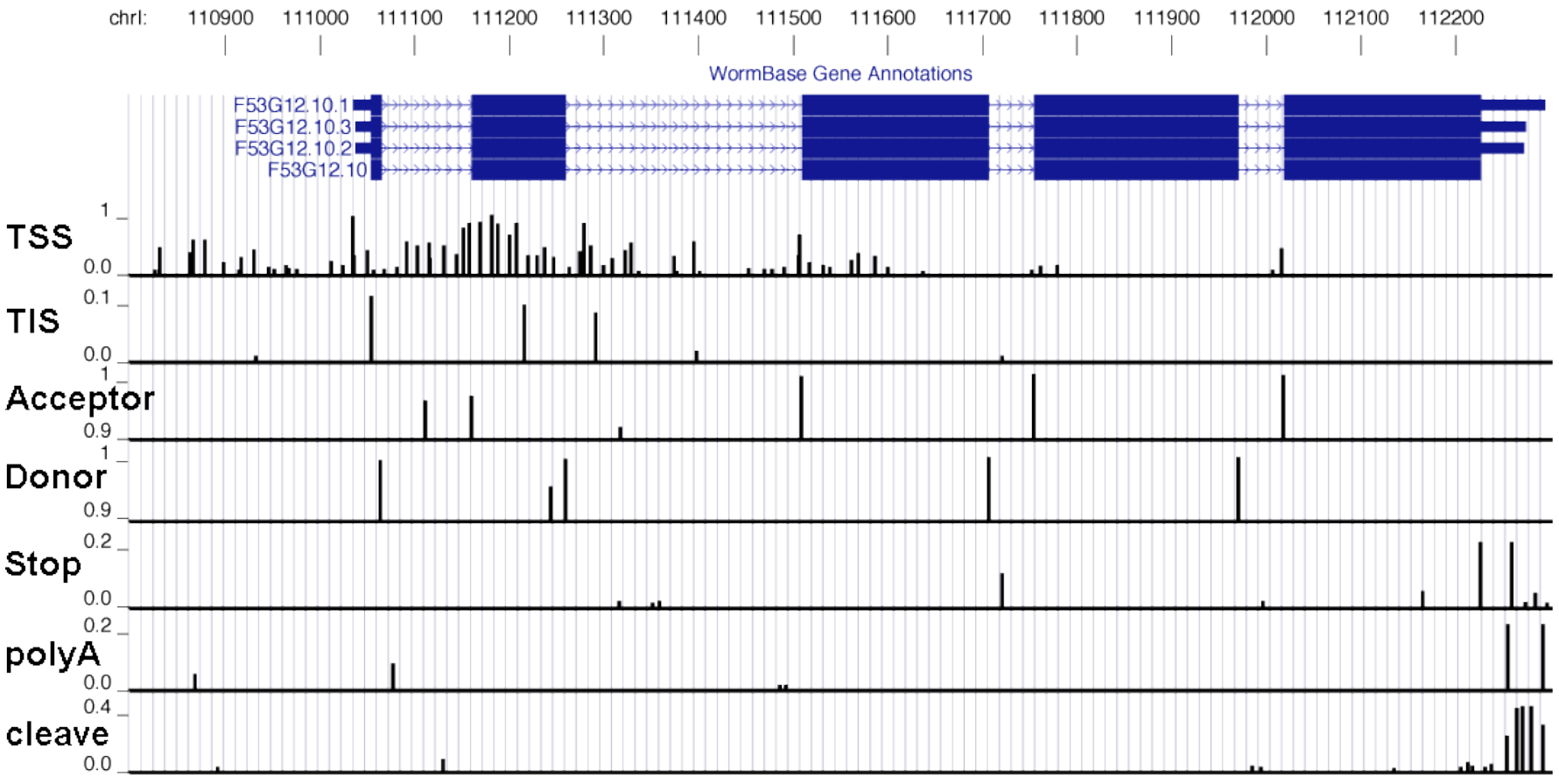
Beauty in Generality



- Transcription Start (Sonnenburg et al., 2006/Down et al. 2002)
- Acceptor Splice Site (Sonnenburg et al., 2007/Baten et al. 2006)
- Donor Splice Site (Sonnenburg et al., 2007/Baten et al. 2006)
- Alternative Splicing (Rätsch, Sonnenburg et al., 2005/-)
- Transsplicing (Schweikert, Sonnenburg et al., 2009/-)
- Translation Initiation (Sonnenburg et al., 2008/Saeys et al., 2007)

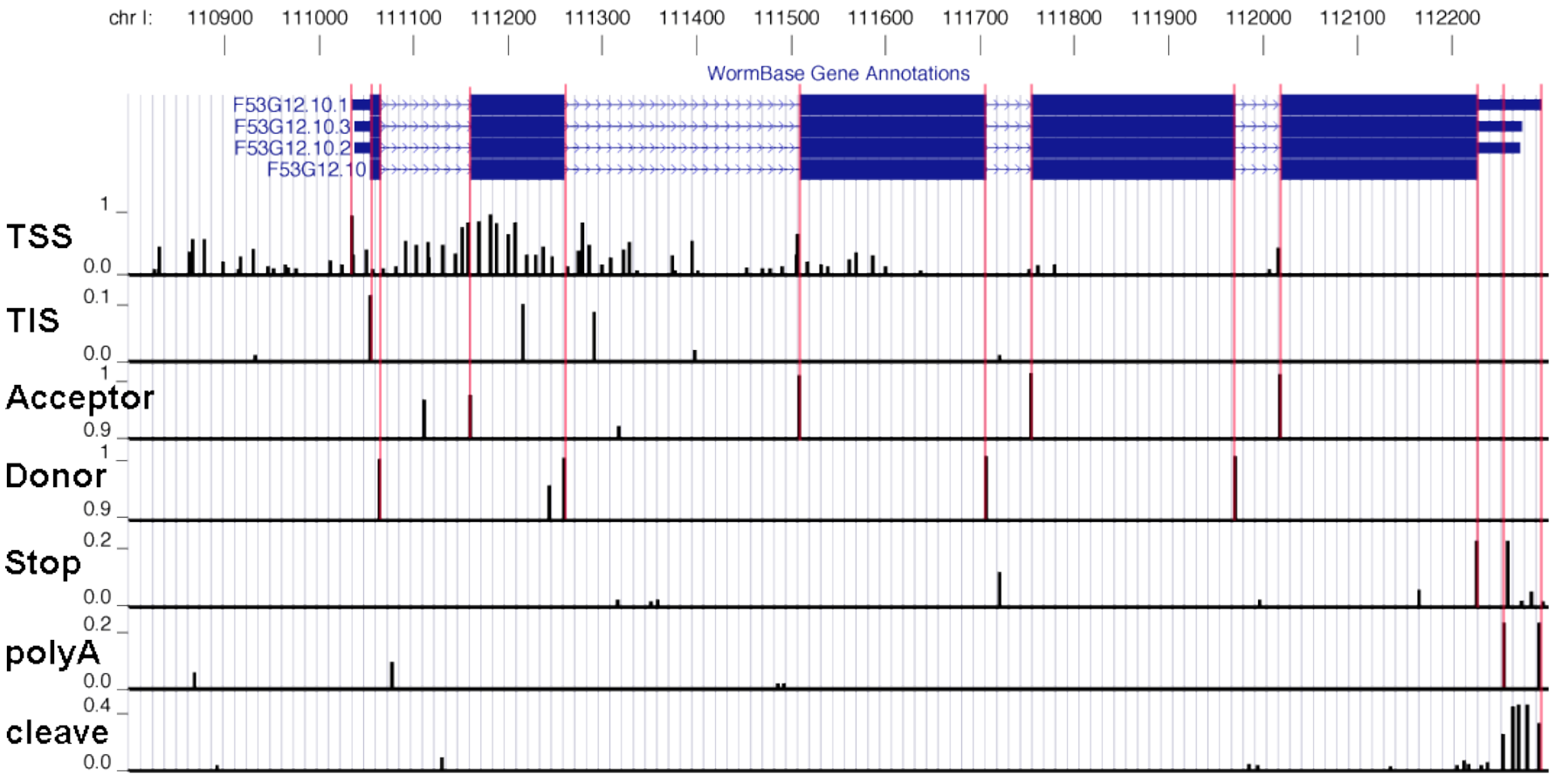
SVM with string kernel often most accurate method.

Individual Signal Predictions in UCSC Browser



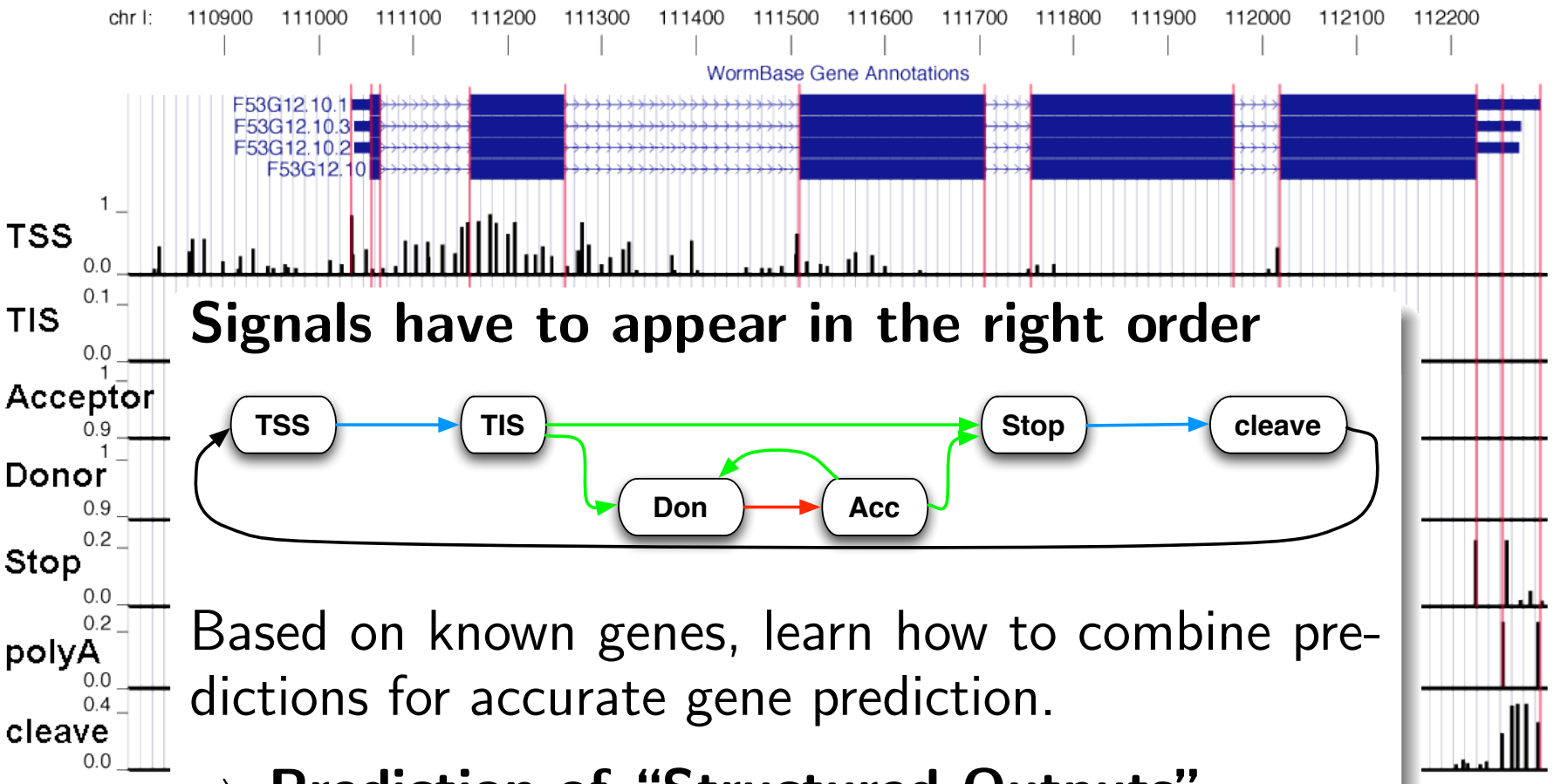
(Raetsch, Sonnenburg et al. 2007, Schweikert et al. 2009)

Individual Signal Predictions in UCSC Browser



(Raetsch, Sonnenburg et al. 2007, Schweikert et al. 2009)

Individual Signal Predictions in UCSC Browser



Signals have to appear in the right order

Based on known genes, learn how to combine predictions for accurate gene prediction.

⇒ **Prediction of "Structured Outputs"**

(Raetsch, Sonnenburg et al. 2007, Schweikert et al. 2009)

Results using mGene (Schweikert et al. 2009)

- Most accurate *ab initio* method in the nGASP genome annotation challenge for *C. elegans*
- Validation of gene predictions for *C. elegans*:

	No. of genes	No. of genes analyzed	Frac. of genes w/ expression
New genes	2,197	57	$\approx 42\%$
Missing unconf. genes	205	24	$\approx 8\%$

- Annotation of other nematode genomes:

Genome	Genome size [Mbp]	No. of genes	No. exons/gene (mean)	mGene accuracy	best other accuracy
C. remanei	235.94	31503	5.7	**96.6%**	93.8%
C. japonica	266.90	20121	5.3	**93.3%**	88.7%
C. brenneri	453.09	41129	5.4	**93.1%**	87.8%
C. briggsae	108.48	22542	6.0	**87.0%**	82.0%

- Web service for *ab initio* gene predictions

Results using mGene (Schweikert et al. 2009)

- Most accurate *ab initio* method in the nGASP genome annotation challenge for *C. elegans*
- Validation of gene predictions for *C. elegans*:

	No. of genes	No. of genes analyzed	Frac. of genes w/ expression
New genes	2,197	57	$\approx 42\%$
Missing unconf. genes	205	24	$\approx 8\%$

- Annotation of other nematode genomes:

Genome	Genome size [Mbp]	No. of genes	No. exons/gene (mean)	mGene accuracy	best other accuracy
C. remanei	235.94	31503	5.7	**96.6%**	93.8%
C. japonica	266.90	20121	5.3	**93.3%**	88.7%
C. brenneri	453.09	41129	5.4	**93.1%**	87.8%
C. briggsae	108.48	22542	6.0	**87.0%**	82.0%

- Web service for *ab initio* gene predictions

Results using mGene (Schweikert et al. 2009)

- Most accurate *ab initio* method in the nGASP genome annotation challenge for *C. elegans*
- Validation of gene predictions for *C. elegans*:

	No. of genes	No. of genes analyzed	Frac. of genes w/ expression
New genes	2,197	57	$\approx 42\%$
Missing unconf. genes	205	24	$\approx 8\%$

- Annotation of other nematode genomes:

Genome	Genome size [Mbp]	No. of genes	No. exons/gene (mean)	mGene accuracy	best other accuracy
C. remanei	235.94	31503	5.7	**96.6%**	93.8%
C. japonica	266.90	20121	5.3	**93.3%**	88.7%
C. brenneri	453.09	41129	5.4	**93.1%**	87.8%
C. briggsae	108.48	22542	6.0	**87.0%**	82.0%

- Web service for *ab initio* gene predictions

Results using mGene (Schweikert et al. 2009)

- Most accurate *ab initio* method in the nGASP genome annotation challenge for *C. elegans*
- Validation of gene predictions for *C. elegans*:

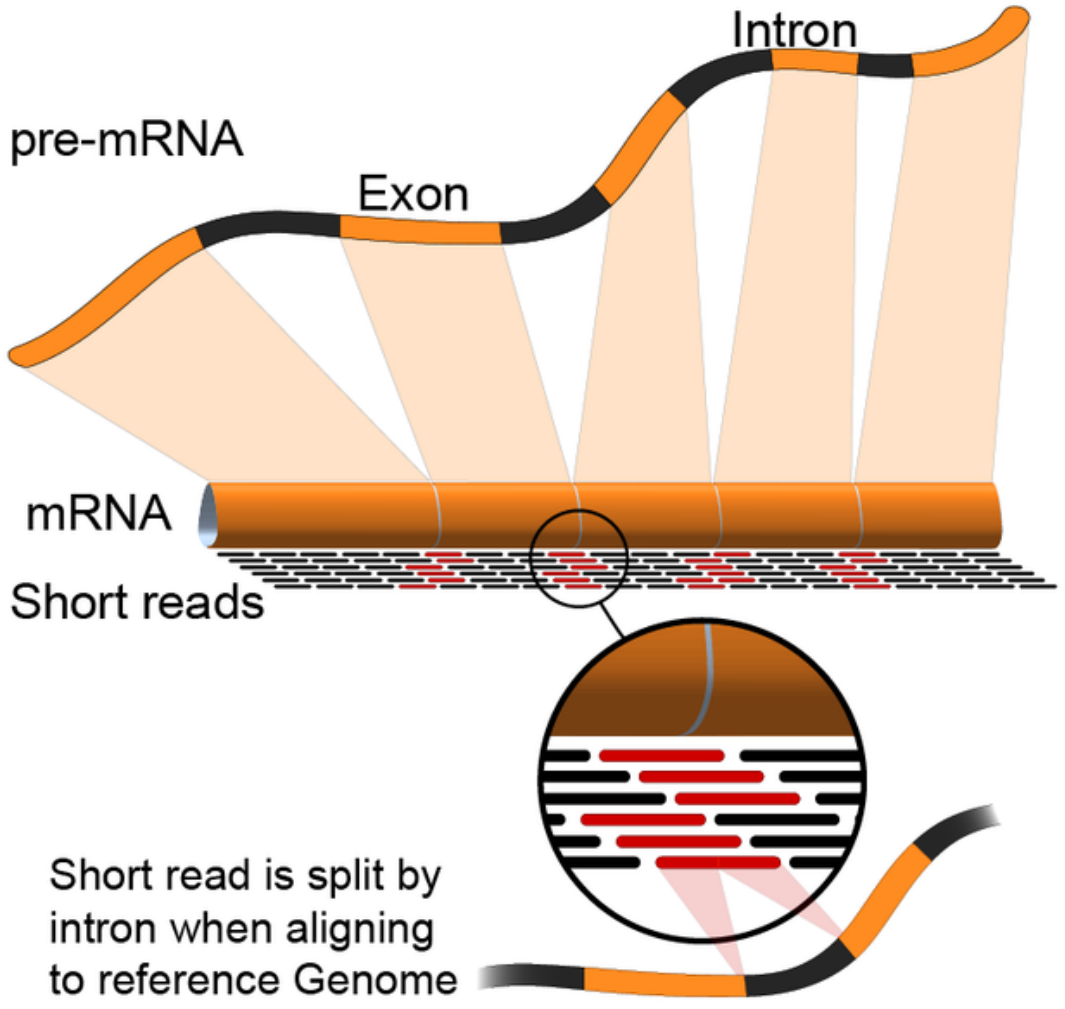
	No. of genes	No. of genes analyzed	Frac. of genes w/ expression
New genes	2,197	57	$\approx 42\%$
Missing unconf. genes	205	24	$\approx 8\%$

- Annotation of other nematode genomes:

Genome	Genome size [Mbp]	No. of genes	No. exons/gene (mean)	mGene accuracy	best other accuracy
C. remanei	235.94	31503	5.7	**96.6%**	93.8%
C. japonica	266.90	20121	5.3	**93.3%**	88.7%
C. brenneri	453.09	41129	5.4	**93.1%**	87.8%
C. briggsae	108.48	22542	6.0	**87.0%**	82.0%

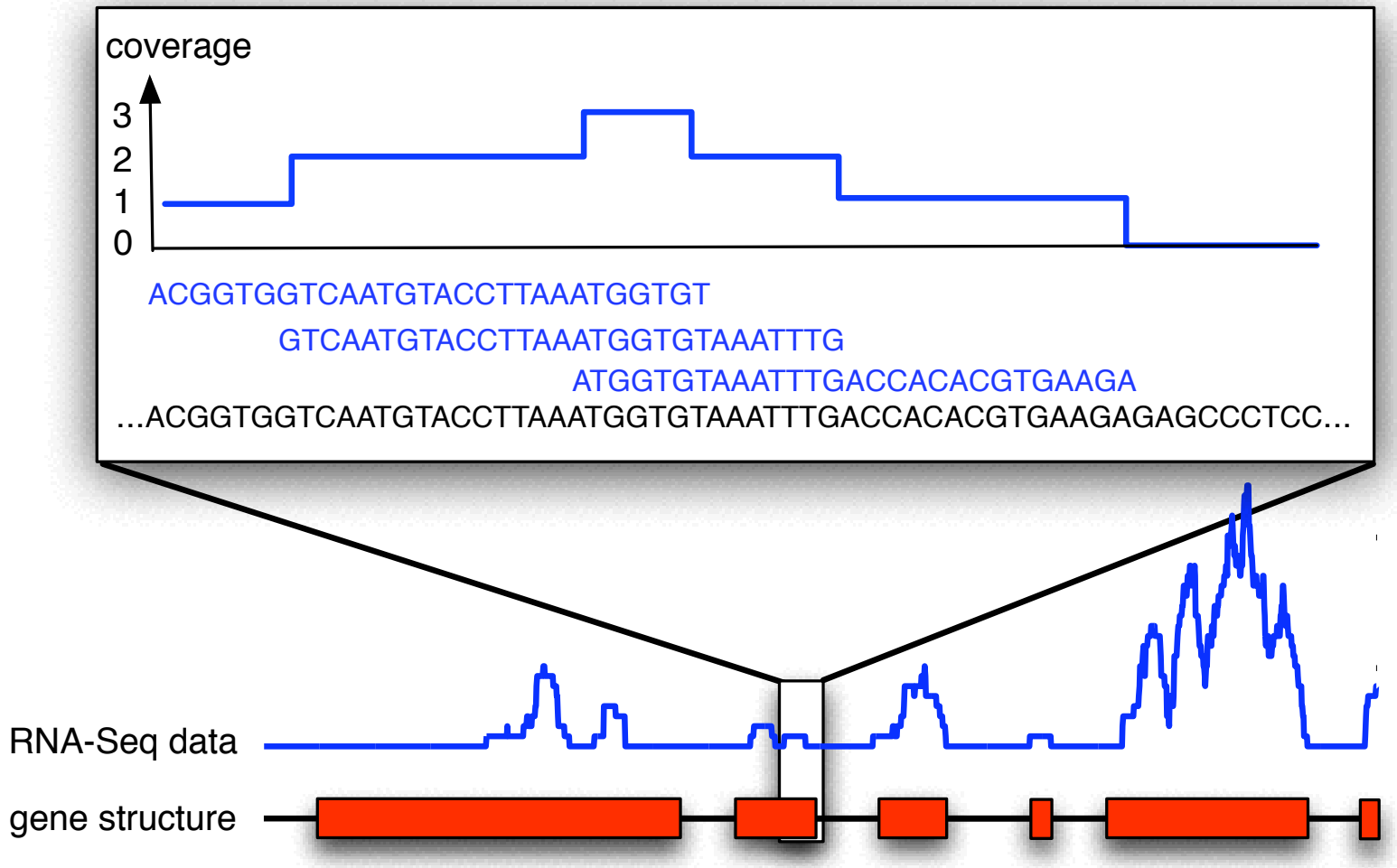
- Web service for *ab initio* gene predictions

mRNA Deep Sequencing



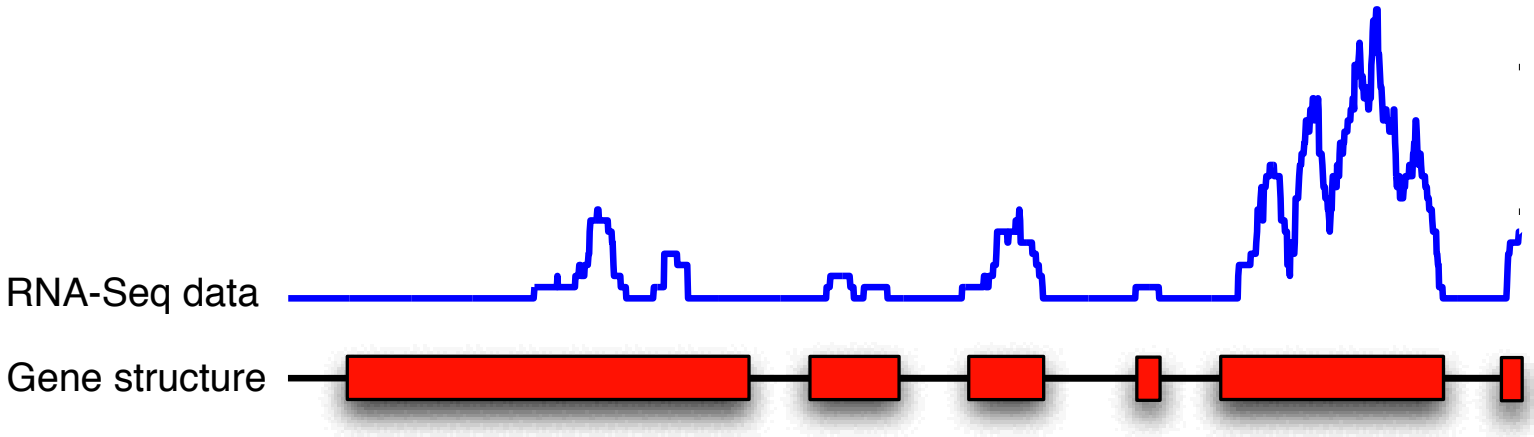
[Wikipedia]

RNA-Seq Read Alignment



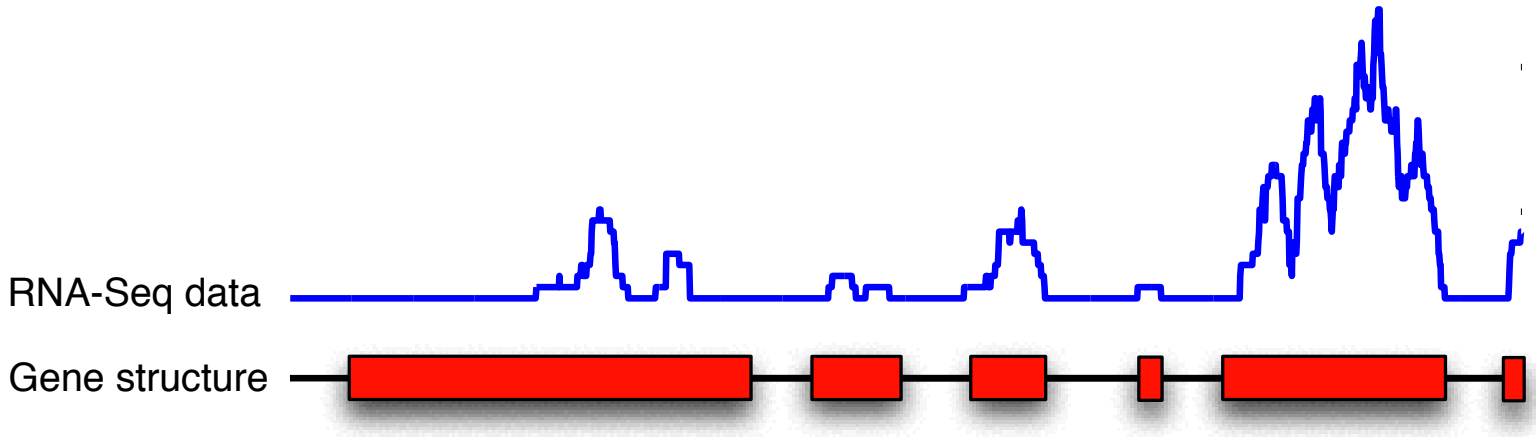
RNA-Seq Read Alignment

```
...GCAAACCAGTGACCTGACTACTACGTCGTAACGTACACGGTAGCT....CCGTAGAATTGACTGTGTTG...
   GCAAACCAGTGACCTGACTACTACGTCGTAACGTAC
```

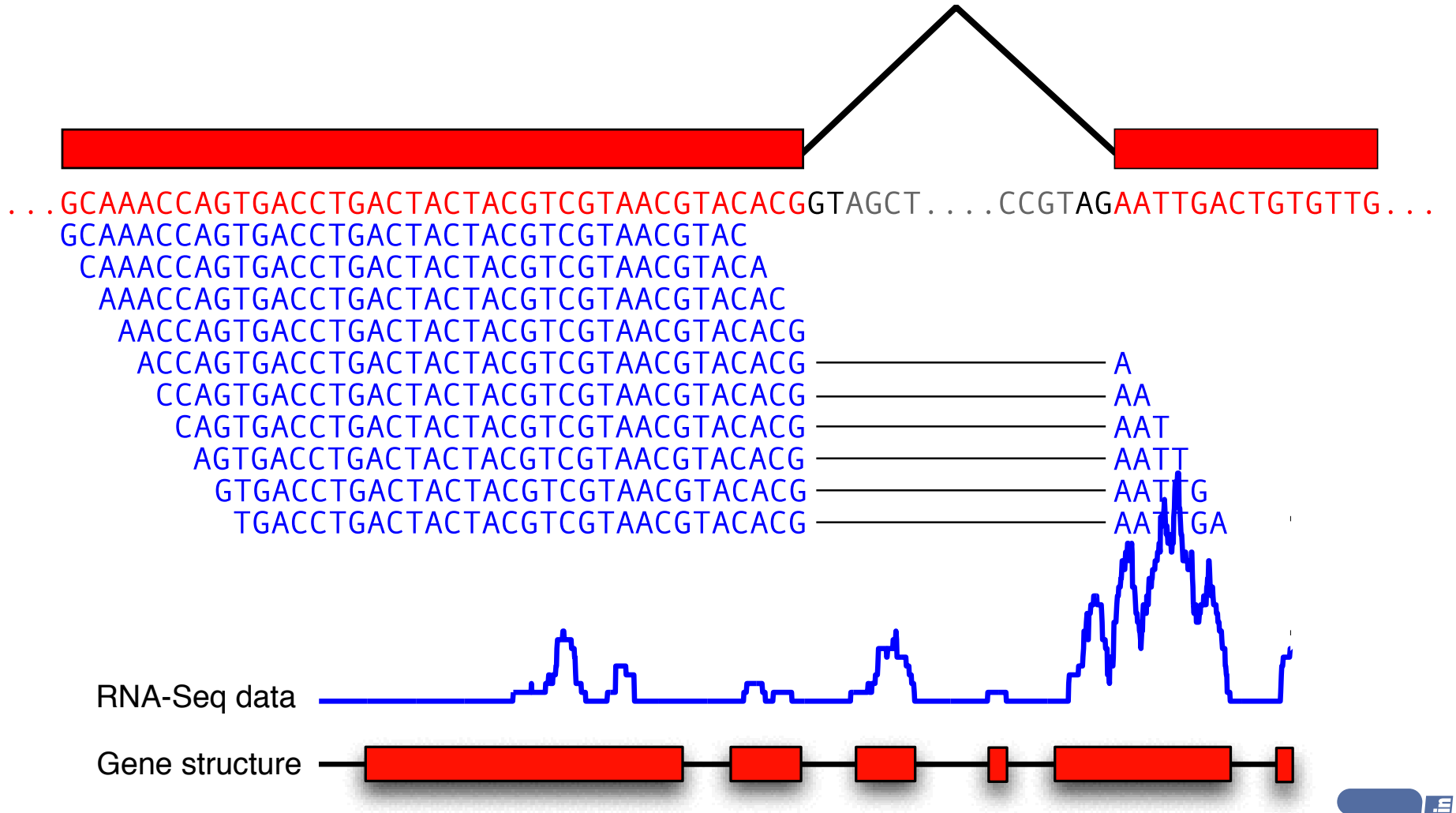


RNA-Seq Read Alignment

```
...GCAAACCAGTGACCTGACTACTACGTCGTAACGTACACGGTAGCT....CCGTAGAATTGACTGTGTTG...
   GCAAACCAGTGACCTGACTACTACGTCGTAACGTAC
    CAAACCAGTGACCTGACTACTACGTCGTAACGTACA
     AAACCAGTGACCTGACTACTACGTCGTAACGTACAC
      AACCAGTGACCTGACTACTACGTCGTAACGTACACG
```



RNA-Seq Read Alignment



Extended Smith-Waterman Algorithm (De Bona et al. 2008)

⇒ **Combine heterogeneous evidence for accurate alignment**

	gap	A	C	G	T	N
gap	0.33	0.3	0.12	0.3	0.3	0.55
A	0.31	0.12	0.12	0.3	0.55	0.33
C	0.44	0.12	0.44	0.3	0.59	0.12
G	0.13	0.85	0.31	0.33	0.51	0.3
T	0.55	0.12	013	0.12	0.11	0.1
N	0.12	0.01	0.3	0.12	0.3	0.01

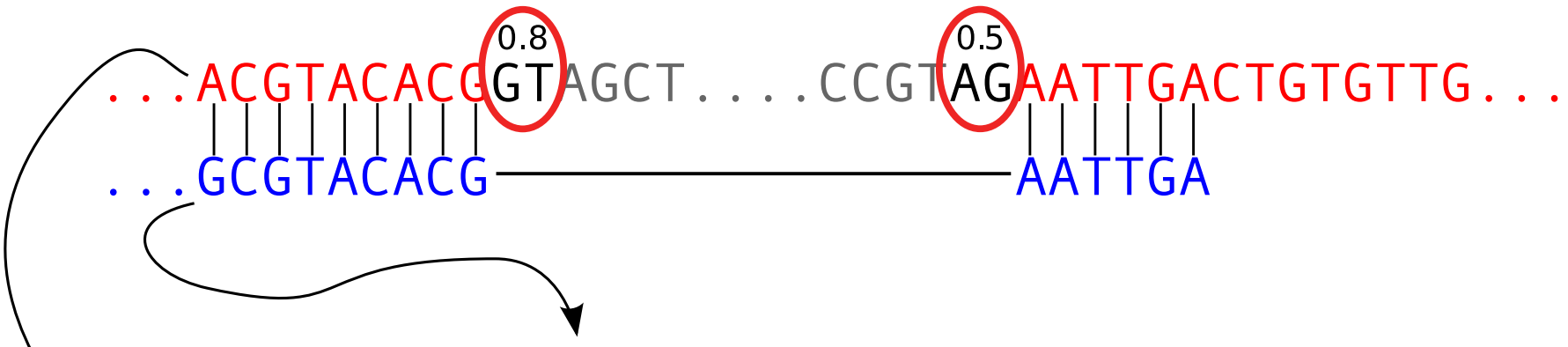
Source of Information

- Sequence matches
- Computational splice site predictions
- Intron length model
- Read quality information

Classical scoring $f : \Sigma \times \Sigma \rightarrow \mathbb{R}$

Extended Smith-Waterman Algorithm (De Bona et al. 2008)

⇒ **Combine heterogeneous evidence for accurate alignment**



	gap	A	C	G	T	N
gap	0.33	0.3	0.12	0.3	0.3	0.55
A	0.31	0.12	0.12	0.3	0.55	0.33
C	0.44	0.12	0.44	0.3	0.59	0.12
G	0.13	0.85	0.31	0.33	0.51	0.3
T	0.55	0.12	013	0.12	0.11	0.1
N	0.12	0.01	0.3	0.12	0.3	0.01

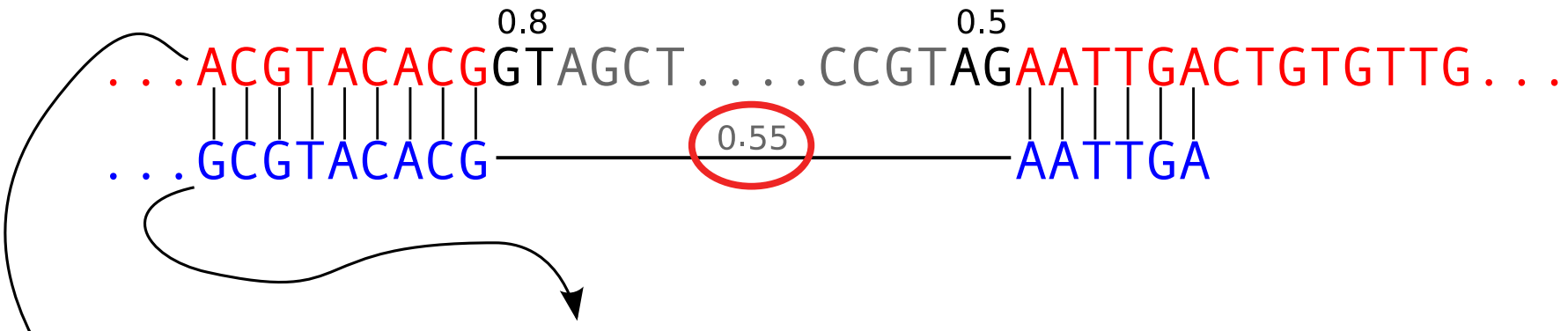
Source of Information

- Sequence matches
- Computational splice site predictions
- Intron length model
- Read quality information

Classical scoring $f : \Sigma \times \Sigma \rightarrow \mathbb{R}$

Extended Smith-Waterman Algorithm (De Bona et al. 2008)

⇒ **Combine heterogeneous evidence for accurate alignment**



	gap	A	C	G	T	N
gap	0.33	0.3	0.12	0.3	0.3	0.55
A	0.31	0.12	0.12	0.3	0.55	0.33
C	0.44	0.12	0.44	0.3	0.59	0.12
G	0.13	0.85	0.31	0.33	0.51	0.3
T	0.55	0.12	013	0.12	0.11	0.1
N	0.12	0.01	0.3	0.12	0.3	0.01

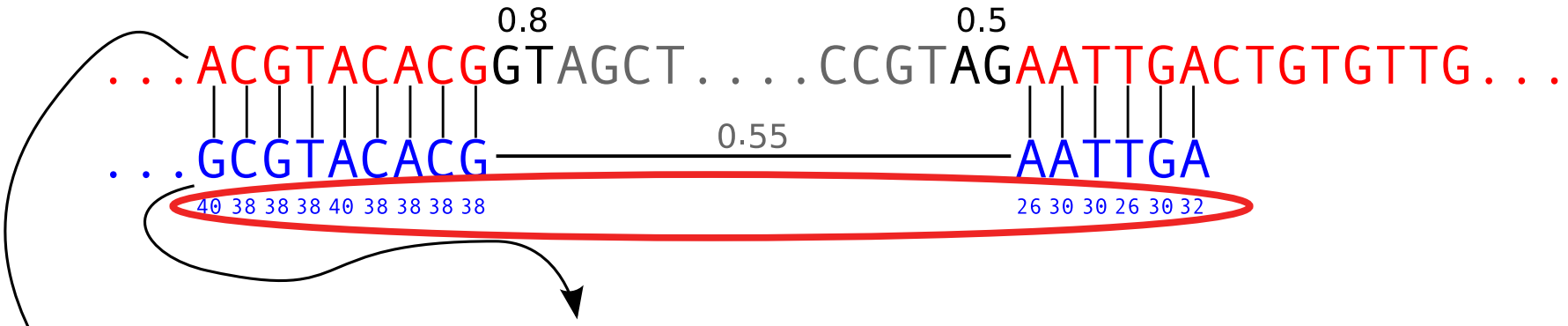
Source of Information

- Sequence matches
- Computational splice site predictions
- Intron length model
- Read quality information

Classical scoring $f : \Sigma \times \Sigma \rightarrow \mathbb{R}$

Extended Smith-Waterman Algorithm (De Bona et al. 2008)

⇒ **Combine heterogeneous evidence for accurate alignment**



	gap	A	C	G	T	N
gap	0.33	$f_{-A}(\cdot)$	$f_{-C}(\cdot)$	$f_{-G}(\cdot)$	$f_{-T}(\cdot)$	$f_{-N}(\cdot)$
A	0.31	$f_{AA}(\cdot)$	$f_{AC}(\cdot)$	$f_{AG}(\cdot)$	$f_{AT}(\cdot)$	$f_{AN}(\cdot)$
C	0.44	$f_{CA}(\cdot)$	$f_{CC}(\cdot)$	$f_{CG}(\cdot)$	$f_{CT}(\cdot)$	$f_{CN}(\cdot)$
G	0.13	$f_{GA}(\cdot)$	$f_{GC}(\cdot)$	$f_{GG}(\cdot)$	$f_{GT}(\cdot)$	$f_{GN}(\cdot)$
T	0.55	$f_{TA}(\cdot)$	$f_{TC}(\cdot)$	$f_{TG}(\cdot)$	$f_{TT}(\cdot)$	$f_{TN}(\cdot)$
N	0.12	$f_{NA}(\cdot)$	$f_{NC}(\cdot)$	$f_{NG}(\cdot)$	$f_{NT}(\cdot)$	$f_{NN}(\cdot)$

Source of Information

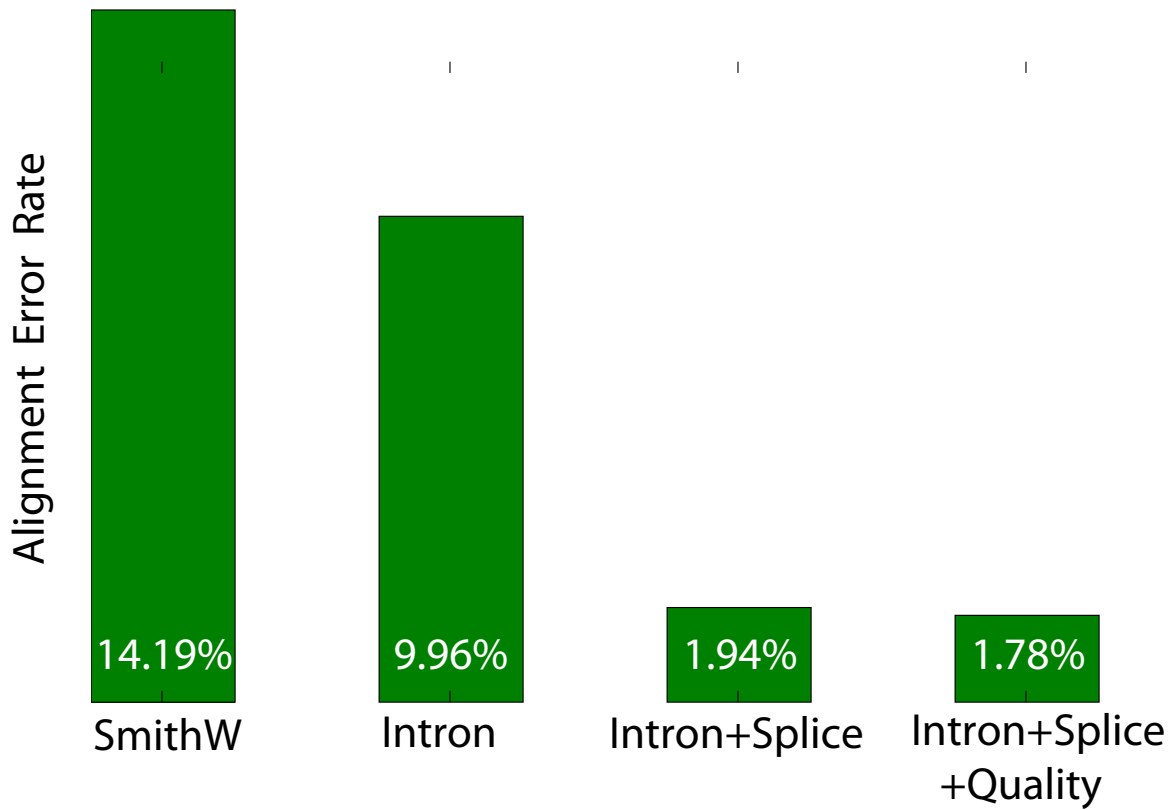
- Sequence matches
- Computational splice site predictions
- Intron length model
- Read quality information

Quality scoring $f : (\Sigma \times \mathbb{R}) \times \Sigma \rightarrow \mathbb{R}$ (De Bona et al. 2008)

QPalma's Accurate Alignments (De Bona et al. 2008)

Generate set of artificially spliced reads

- Genomic reads with quality information
- Genome annotation for artificially splicing the reads
- Use 10,000 reads for training and 30,000 for testing



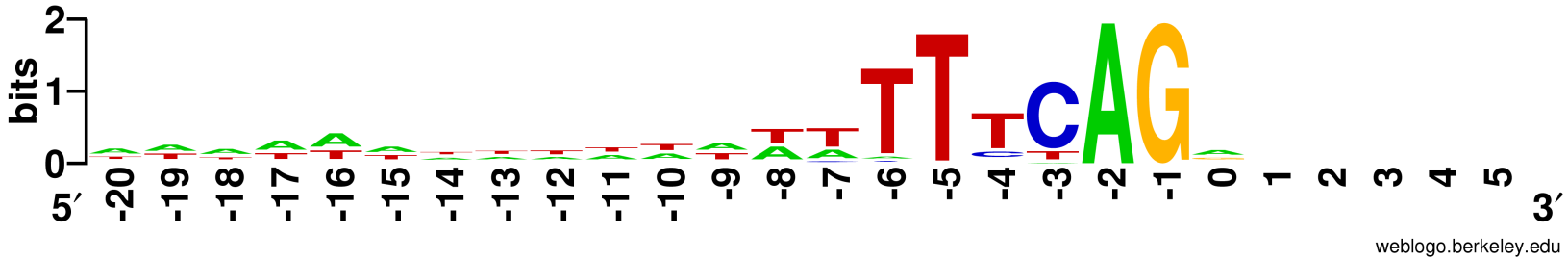
Outline

1. Introduction
   - Genomic Signals
2. Sequence Classification
   - Support Vector Machines
3. Support Vector Machines
4. Applications
   - Splice Site Recognition
   - TSS Recognition
   - Gene Finding - mGene
   - Aligning Short Reads - QPALMA
5. Explanation and Visualization
   - Introduction
   - Definition
   - Applications
6. Summary

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

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

$\alpha_1 \cdot$ AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG
$\alpha_2 \cdot$ AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT
$\alpha_3 \cdot$ CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC
$\vdots$ $\vdots$
$\alpha_N \cdot$ TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC

But we are interested in weights over features.

SVM Scoring Function

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

k-mer	pos. 1	pos. 2	pos. 3	pos. 4	$\cdots$
A	+0.1	-0.3	-0.2	+0.2	$\cdots$
C	0.0	-0.1	+2.4	-0.2	$\cdots$
G	+0.1	-0.7	0.0	-0.5	$\cdots$
T	-0.2	-0.2	0.1	+0.5	$\cdots$
AA	+0.1	-0.3	+0.1	0.0	$\cdots$
AC	+0.2	0.0	-0.2	+0.2	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
TT	0.0	-0.1	+1.7	-0.2	$\cdots$
AAA	+0.1	0.0	0.0	+0.1	$\cdots$
AAC	0.0	-0.1	+1.2	-0.2	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
TTT	+0.2	-0.7	0.0	0.0	$\cdots$

The Scoring System - Examples

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

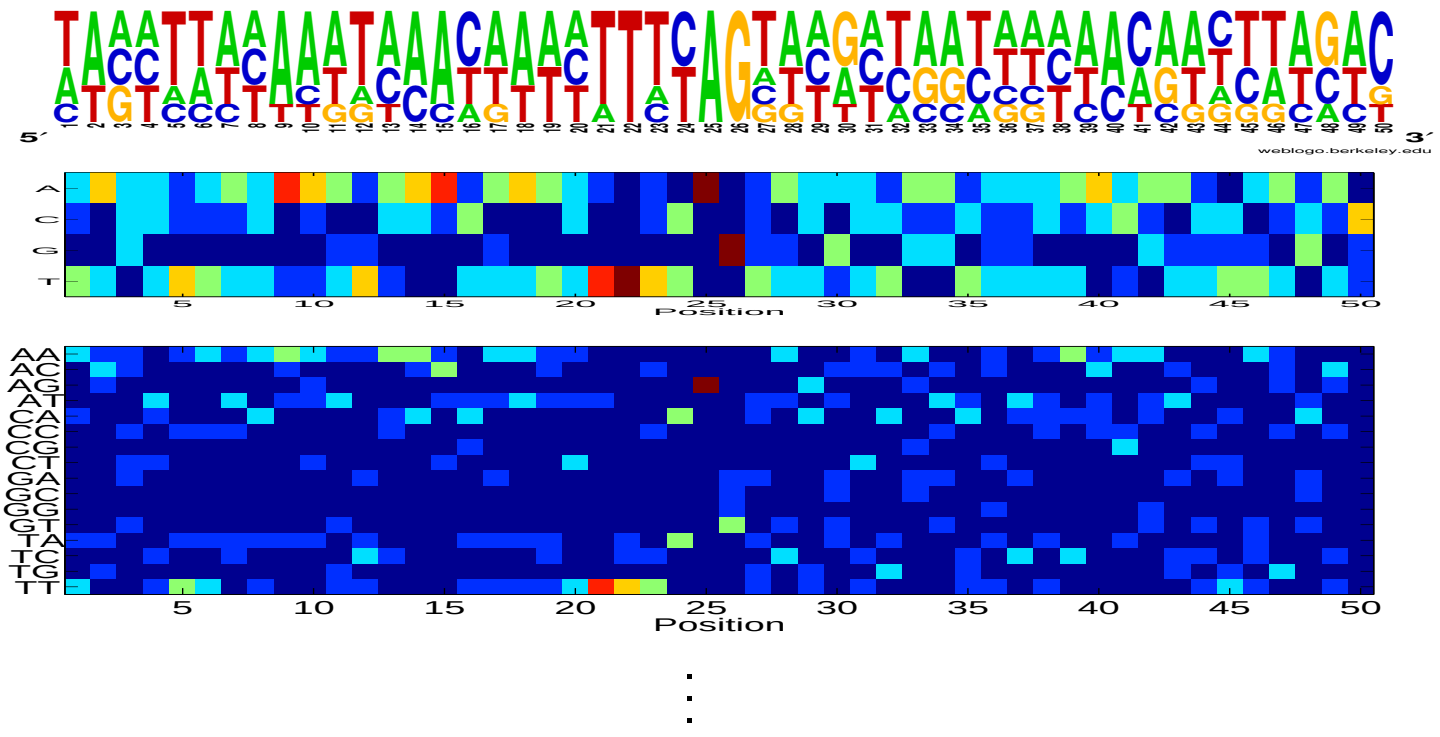
Examples:

- 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)

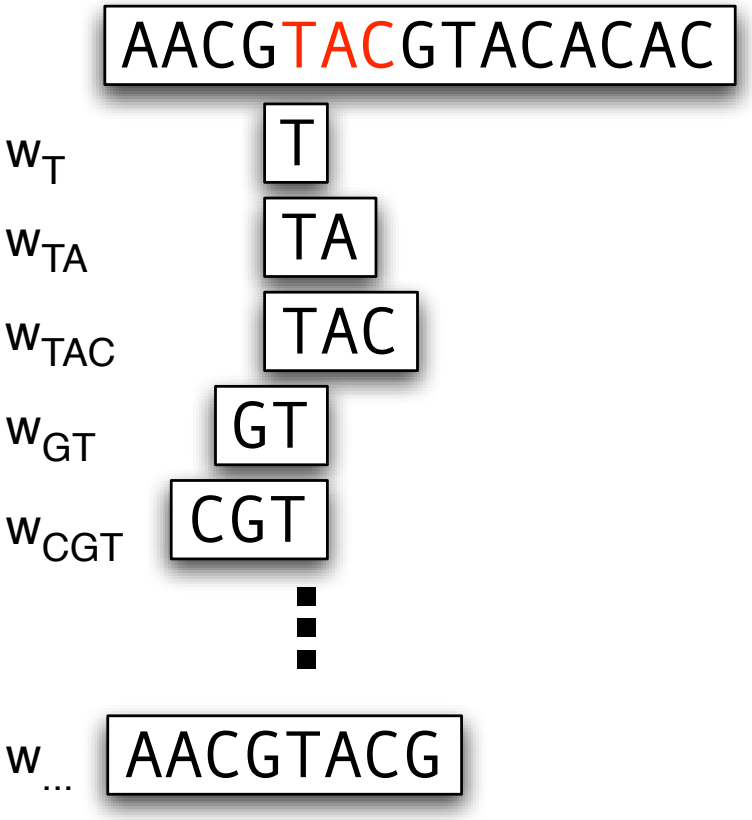
The SVM Weight Vector **w**



- 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 $\mathbb{E}\left[\, s(\mathbf{x}) \mid \mathbf{x}[j] = \mathbf{z} \,\right]$ (**for small k**)

```
AAAAAAAAAATACAAAAAAAAAA
AAAAAAAAAATACAAAAAAAAAC
AAAAAAAAAATACAAAAAAAAAG
          ⋮
TTTTTTTTTTTACTTTTTTTTTT
```

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

POIMs (Sonnenburg et al. (2008))

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

$\Rightarrow$ **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:

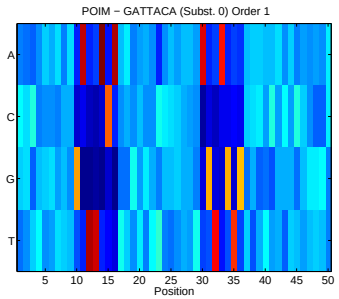
Effort linear in length of input: $\mathcal{O}(LN + L|\Sigma|^k)$

Ranking Features and Condensing Information

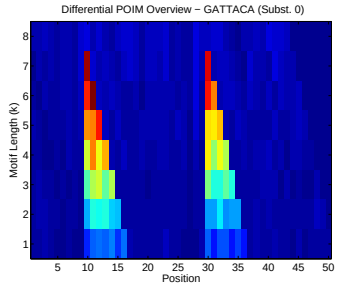
- Obtain highest scoring $\mathbf{z}$ from $Q(\mathbf{z}, i)$ (Enhancer or Silencer)

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

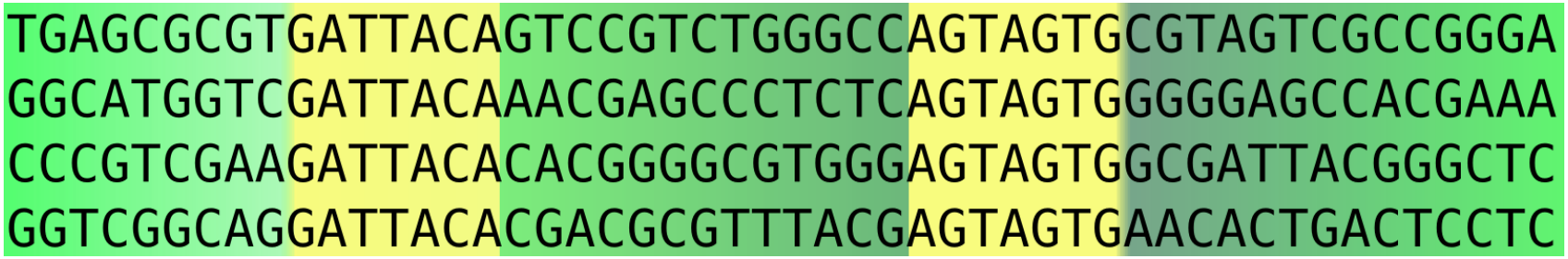
- Visualize POIM as heat map;
  x-axis: position
  y-axis: k-mer
  color: importance



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



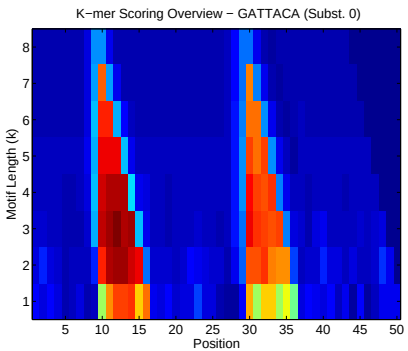
GATTACA and AGTAGTG at Fixed Positions 10 and 30



GATTACA and AGTAGTG at fixed positions 10 and 30

TGAGCGCGTGATTACAGTCCGTCTGGGCCAGTAGTGCGTAGTCGCCGGGA
GGCATGGTCGATTACAAACGAGCCCTCTCAGTAGTGGGGGAGCCACGAAA
CCCGTCGAAGATTACACACGGGGCGTGGGAGTAGTGGCGATTACGGGCTC
GGTCGGCAGGATTACACGACGCGTTTACGAGTAGTGAACACTGACTCCTC

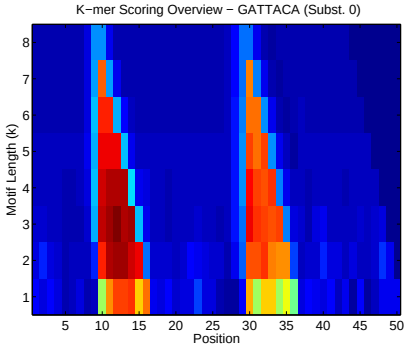
w



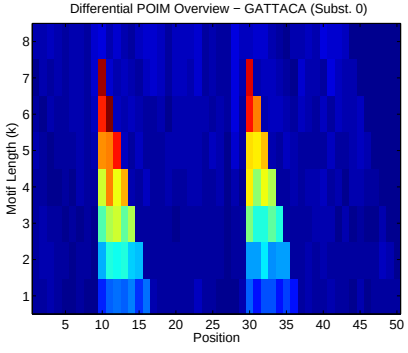
GATTACA and AGTAGTG at fixed positions 10 and 30

TGAGCGCGTGATTACAGTCCGTCTGGGCCAGTAGTGCGTAGTCGCCGGGA
GGCATGGTCGATTACAAACGAGCCCTCTCAGTAGTGGGGGAGCCACGAAA
CCCGTCGAAGATTACACACGGGGCGTGGGAGTAGTGGCGATTACGGGCTC
GGTCGGCAGGATTACACGACGCGTTTACGAGTAGTGAACACTGACTCCTC

w



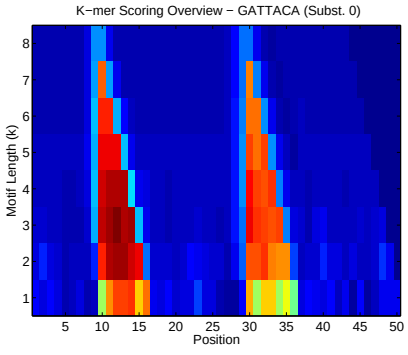
Q

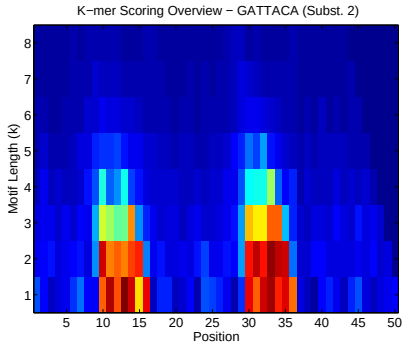


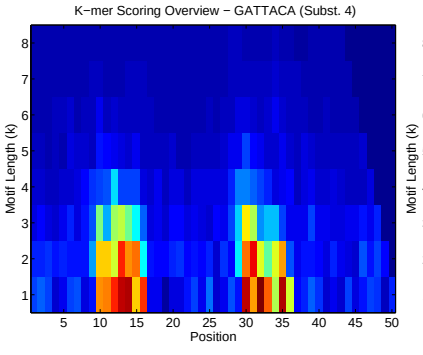
GATTACA and AGTAGTG at fixed positions 10 and 30

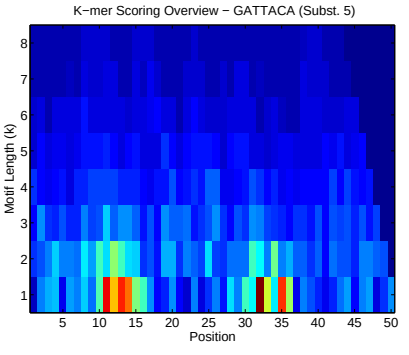
TGAGCGCGTGATTACAGTCCGTCTGGGCCAGTAGTGCGTAGTCGCCGGGA
GGCATGGTCGATTACAAACGAGCCCTCTCAGTAGTGGGGGAGCCACGAAA
CCCGTCGAAGATTACACACGGGGCGTGGGAGTAGTGGCGATTACGGGCTC
GGTCGGCAGGATTACACGACGCGTTTACGAGTAGTGAACACTGACTCCTC

w

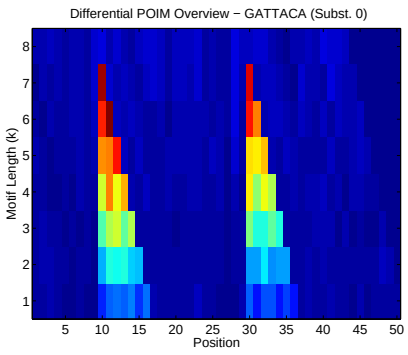


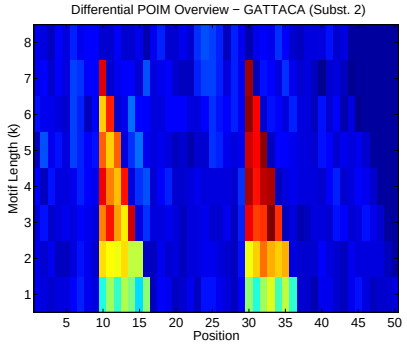


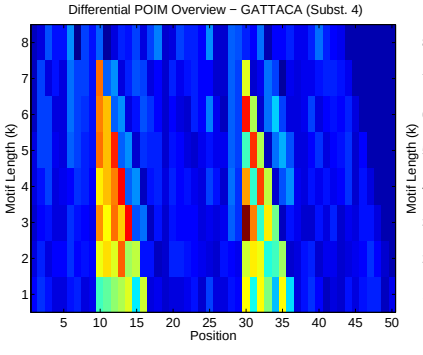


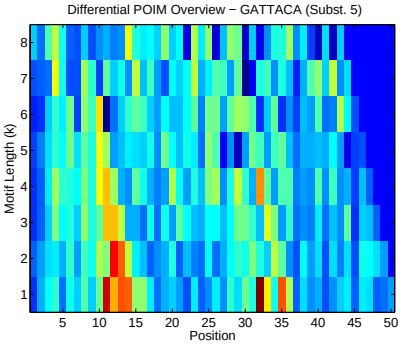


Q

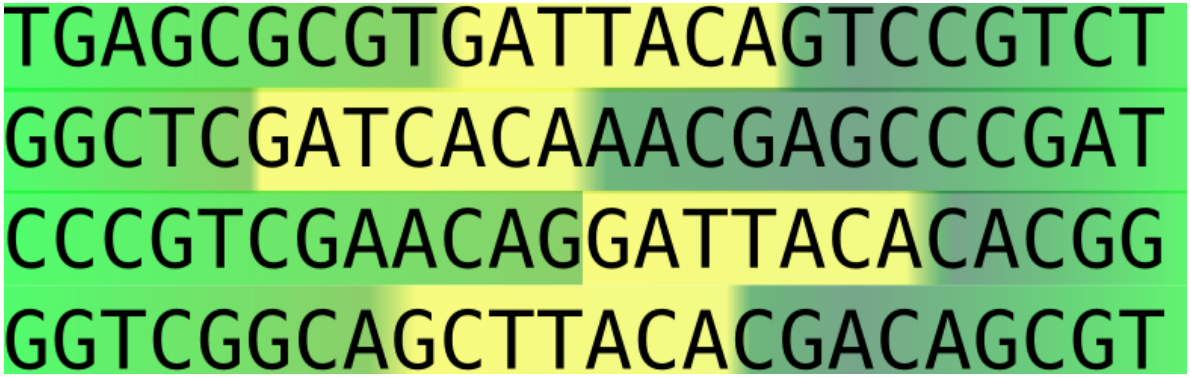






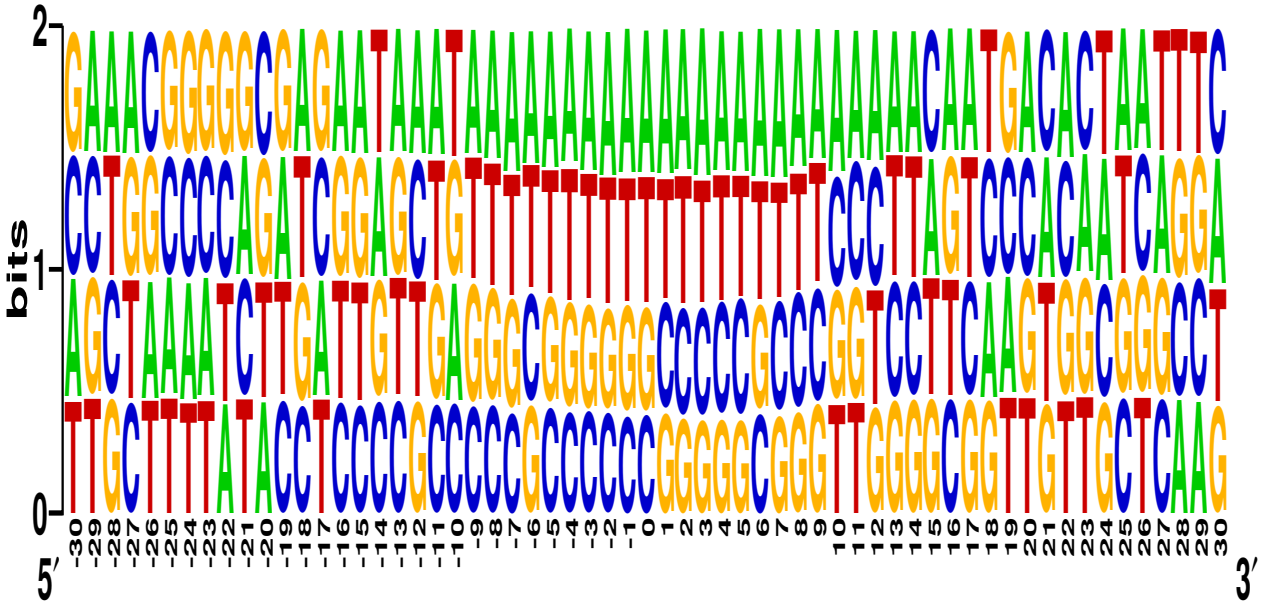


GATTACA at variable positions



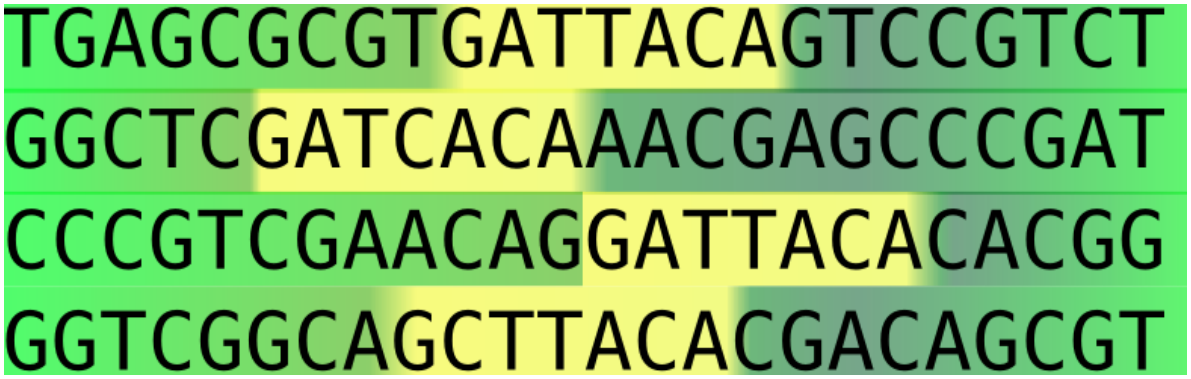
GATTACA at variable positions

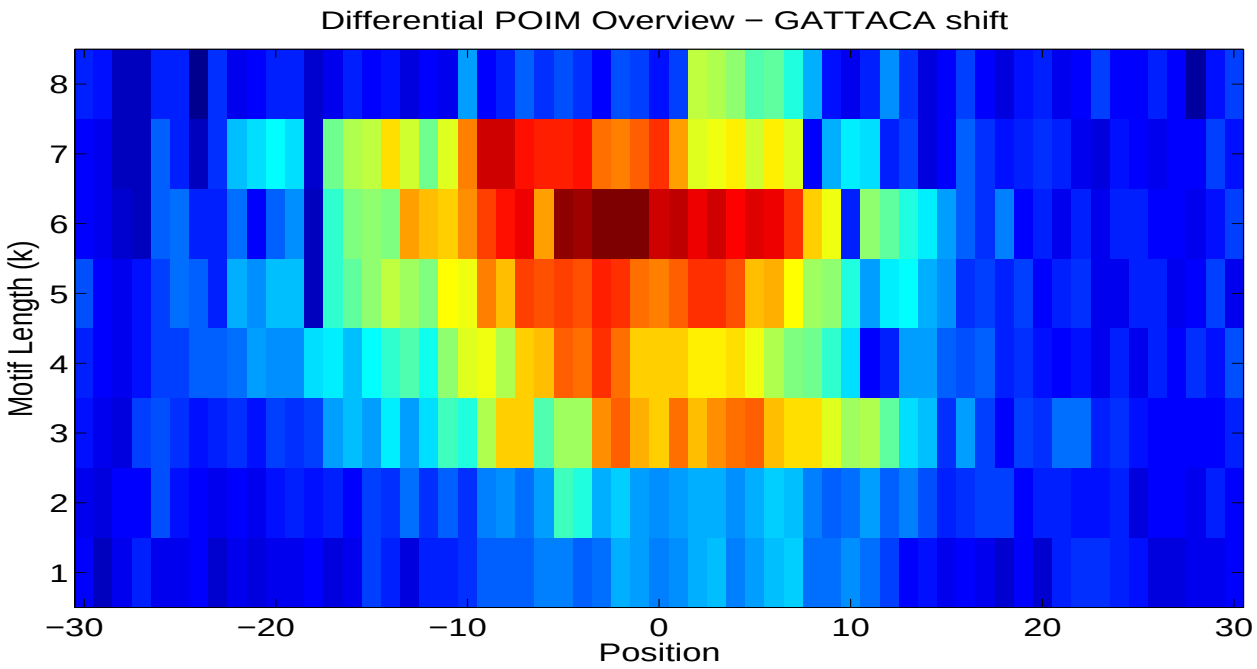
TGAGCGCGTGATTACAGTCCGTCT
GGCTCGATCACAAACGAGCCCGAT
CCCGTCGAACAGGATTACACACGG
GGTCGGCAGCTTACACGACAGCGT



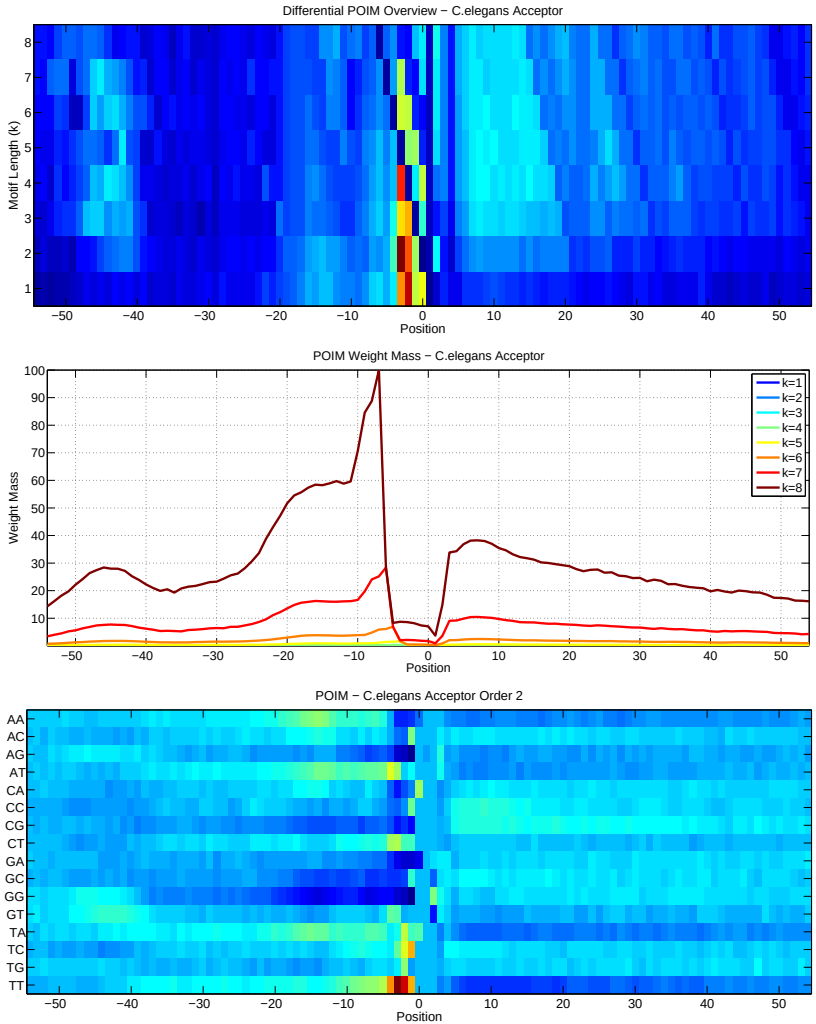
weblogo.berkeley.edu

GATTACA at variable positions



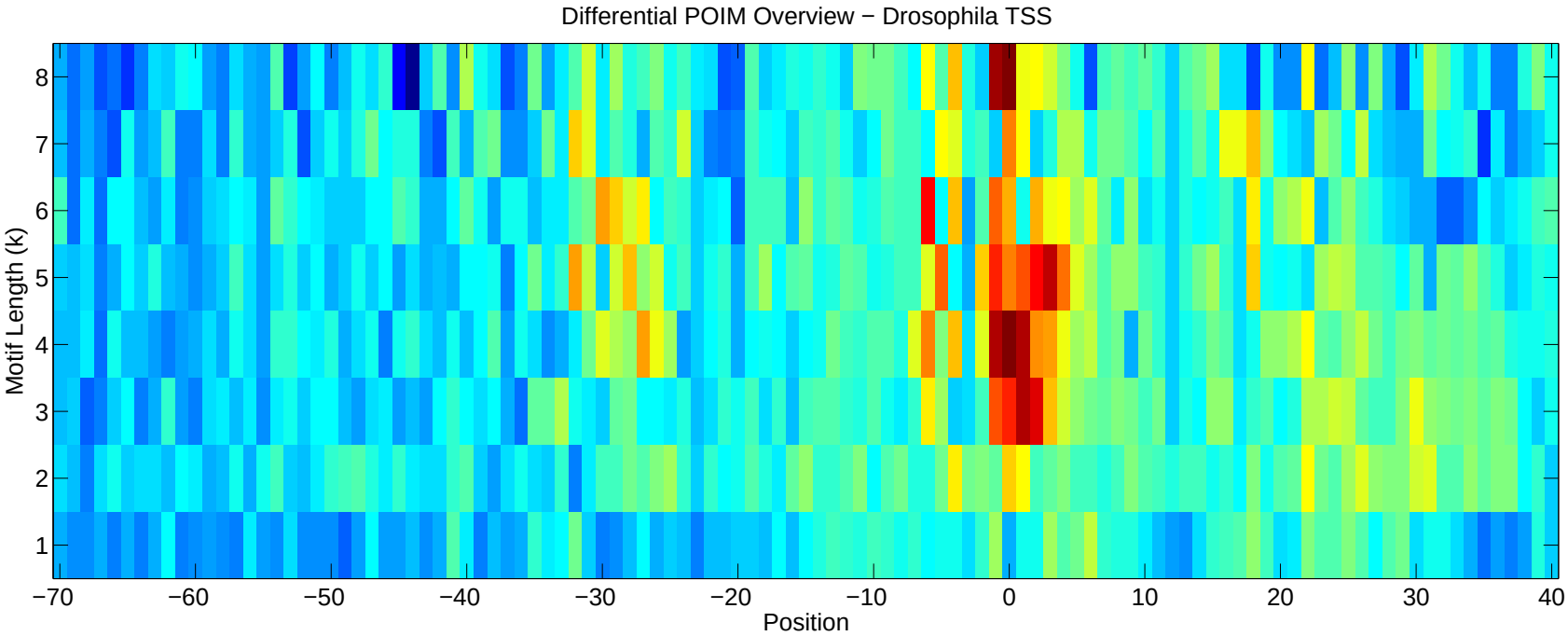


C.elegans Acceptor Splice Site Recognition



- 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

1. Introduction
   - Genomic Signals
2. Sequence Classification
   - Support Vector Machines
3. Support Vector Machines
4. Applications
   - Splice Site Recognition
   - TSS Recognition
   - Gene Finding - mGene
   - Aligning Short Reads - QPALMA
5. Explanation and Visualization
   - Introduction
   - Definition
   - Applications
6. Summary

Conclusions

Support Vector Machines with String Kernels

- General and often state-of-the art signal detectors
- Applicable to genome-sized datasets
- Using POIMs SVMs are interpretable

Software Available

- TSS Detector ARTS `http://mloss.org/software/view/191/`
- Accurate splice detector `http://mloss.org/software/view/192/`
- mGene `http://mgene.org`
- QPalma `http://www.fml.tuebingen.mpg.de/raetsch/suppl/qpalma`
- ML implemented in `http://www.shogun-toolbox.org`

More machine learning software `http://mloss.org`