Introduction to Genomics and Predictive Sequence Analysis

Gunnar Rätsch

Friedrich Miescher Laboratory, Max Planck Society

Tübingen, Germany

Erice Course, September 13, 2010

Discovery of the Nuclein









Discovery of Nuclein:

- from lymphocyte & salmon
- "multi-basic acid" (≥ 4)

"If one . . . wants to assume that a single substance . . . is the specific cause of fertilization, then one should undoubtedly first and foremost consider nuclein" (Miescher, 1874)

Discovery of the Nuclein









Discovery of Nuclein:

- from lymphocyte & salmon
- "multi-basic acid" (≥ 4)

"If one . . . wants to assume that a single substance . . . is the specific cause of fertilization, then one should undoubtedly first and foremost consider nuclein" (Miescher, 1874)

Transcription & RNA Processing in Eucaryotes

- Newly synthesized pre-mRNA is capped.

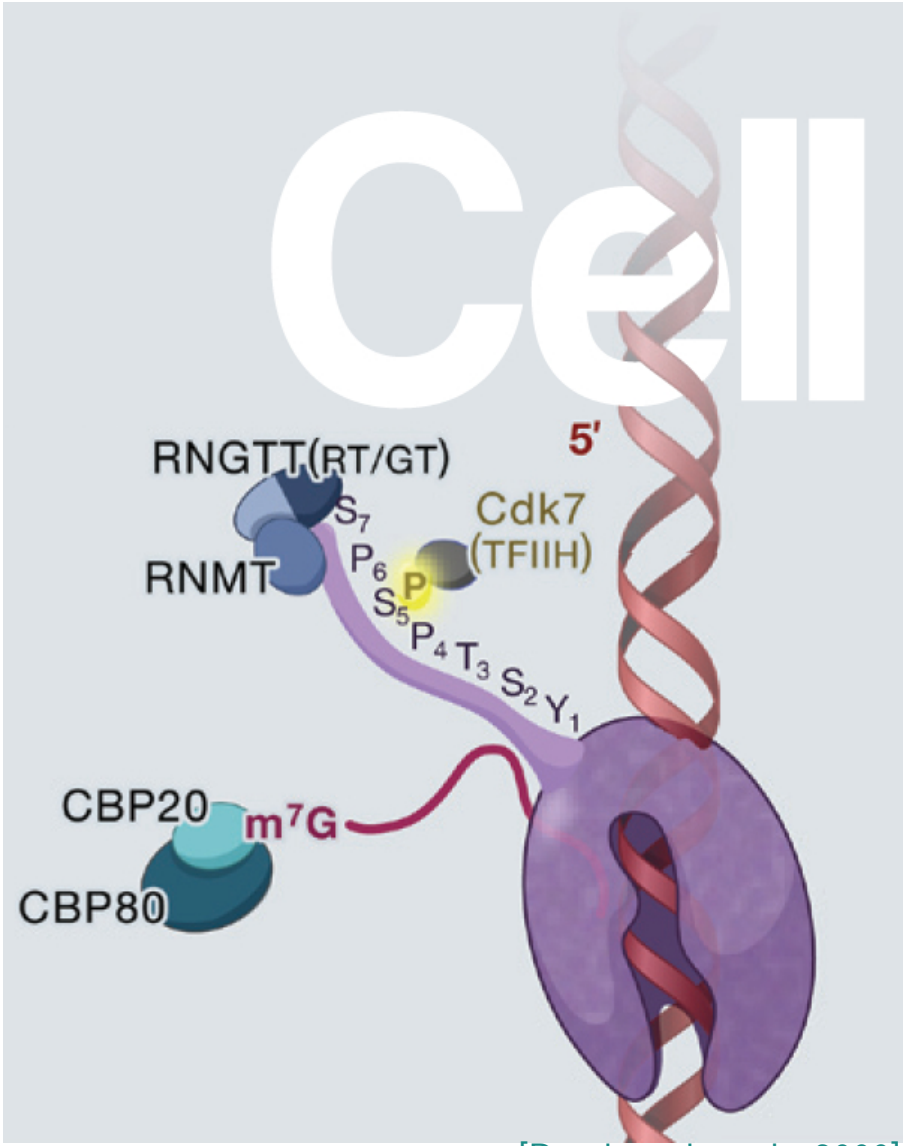
  [CBP20 & CBP80: cap-binding proteins]

- Introns are spliced from pre-mRNA.

  [U1, U2, U4-6: spliceosome

  SF1, U2AF, SR proteins: splicing factors]

- A polyA-tail is added to the 3' terminus of pre-mRNA.



[Bergkessel et al., 2009]

Transcription & RNA Processing in Eucaryotes

- Newly synthesized pre-mRNA is capped.

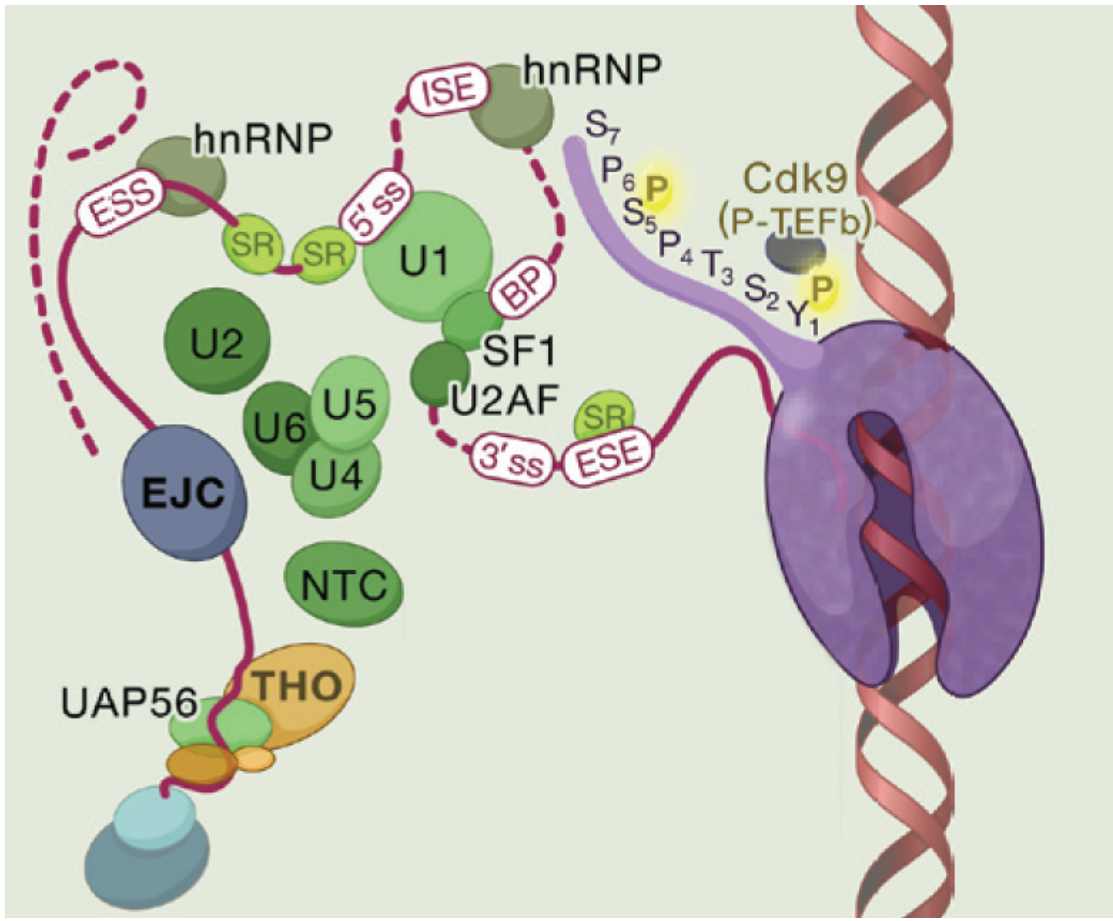
  [CBP20 & CBP80: cap-binding proteins]

- Introns are spliced from pre-mRNA.

  [U1, U2, U4-6: spliceosome

  SF1, U2AF, SR proteins: splicing factors]

- A polyA-tail is added to the 3' terminus of pre-mRNA.



[Bergkessel et al., 2009]

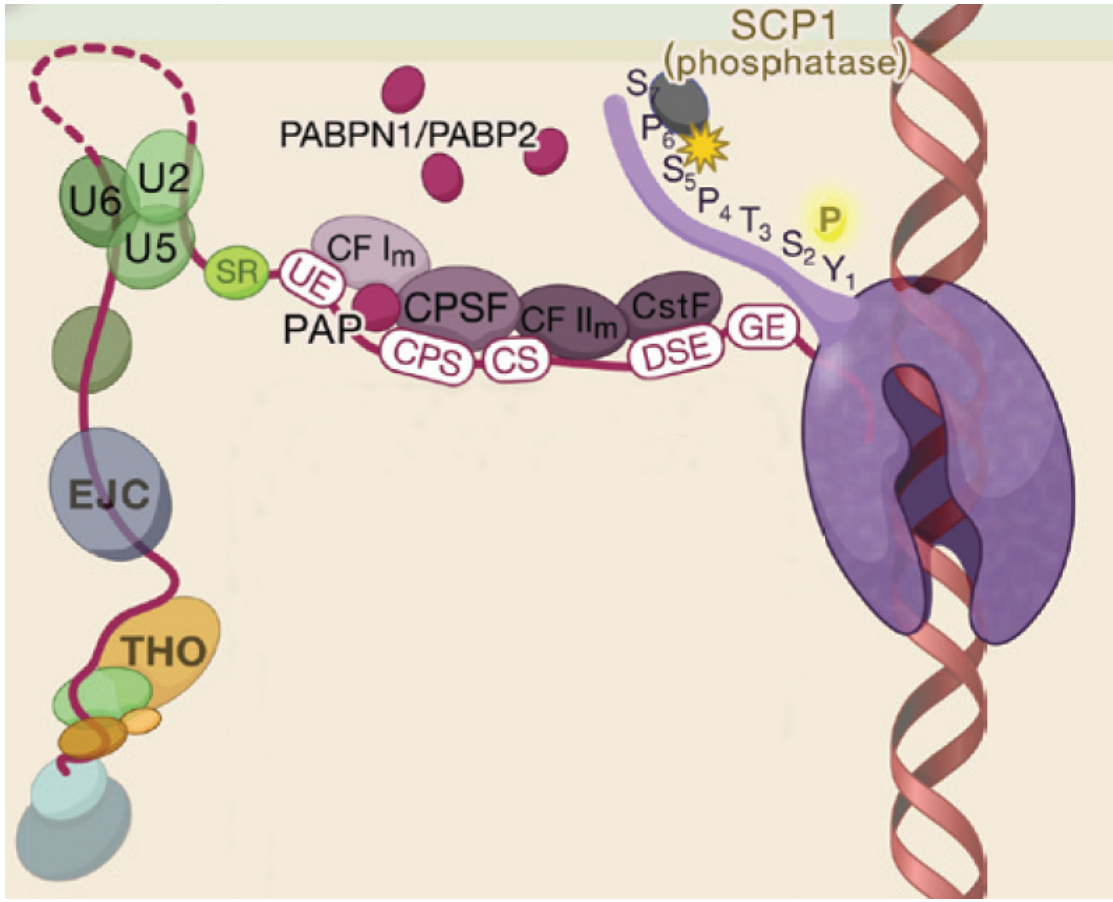
Transcription & RNA Processing in Eucaryotes

- Newly synthesized pre-mRNA is capped.

  [CBP20 & CBP80: cap-binding proteins]
- Introns are spliced from pre-mRNA.

  [U1, U2, U4-6: spliceosome

  SF1, U2AF, SR proteins: splicing factors]
- A polyA-tail is added to the 3' terminus of pre-mRNA.



[Bergkessel et al., 2009]

Overview



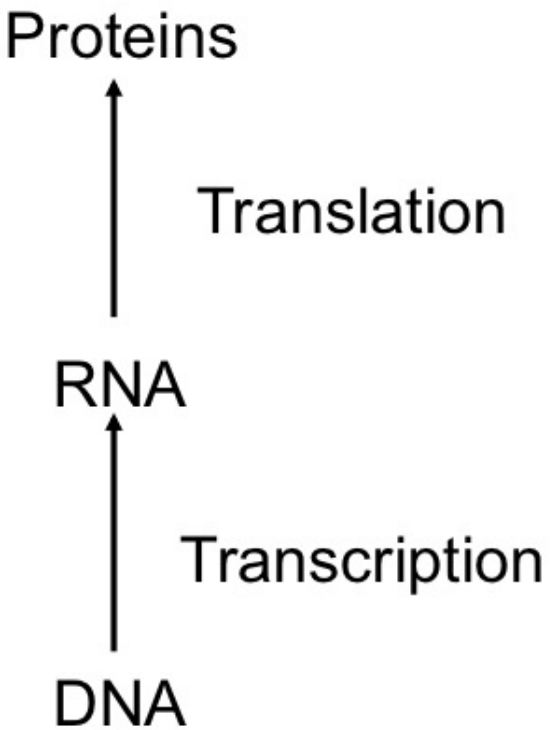
Today:

- History of Genomics and Sequencing
- Challenges in (Functional) Genomics
- Sequence Analysis with SVMs
- Sequence Segmentation with HM-SVMs
- Applications

Tomorrow:

- Algorithms for Deep Sequencing (requires methods from today)

The Central Dogma and Transcriptional Regulation

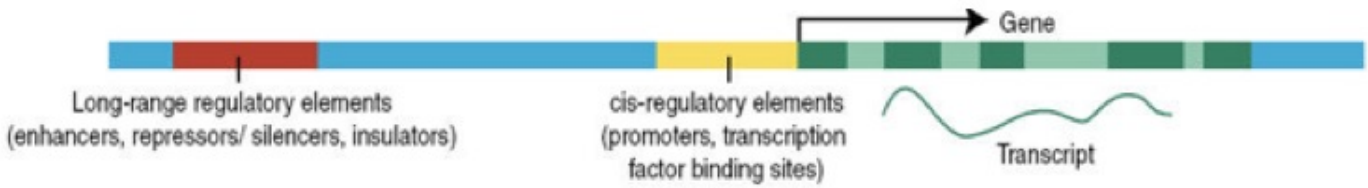


- Transcription is tightly regulated as part of development
  - Also easier to measure than protein levels
- Critical questions:
  - Which genes are turned on and off in a given cell at a given time?
  - What is the expression level of these genes?
  - How is this all encoded in the DNA?

Generalizing by Going Genome-wide



- The cloning of individual genes during the last quarter of the 20th century revealed tantalizing hints to the structure of eukaryotic genes.



- But a comprehensive picture of gene regulation needs to include the entire gene collection for a given organism as well as its intergenic regions, collectively called “genome”.

Sequencing Genomes



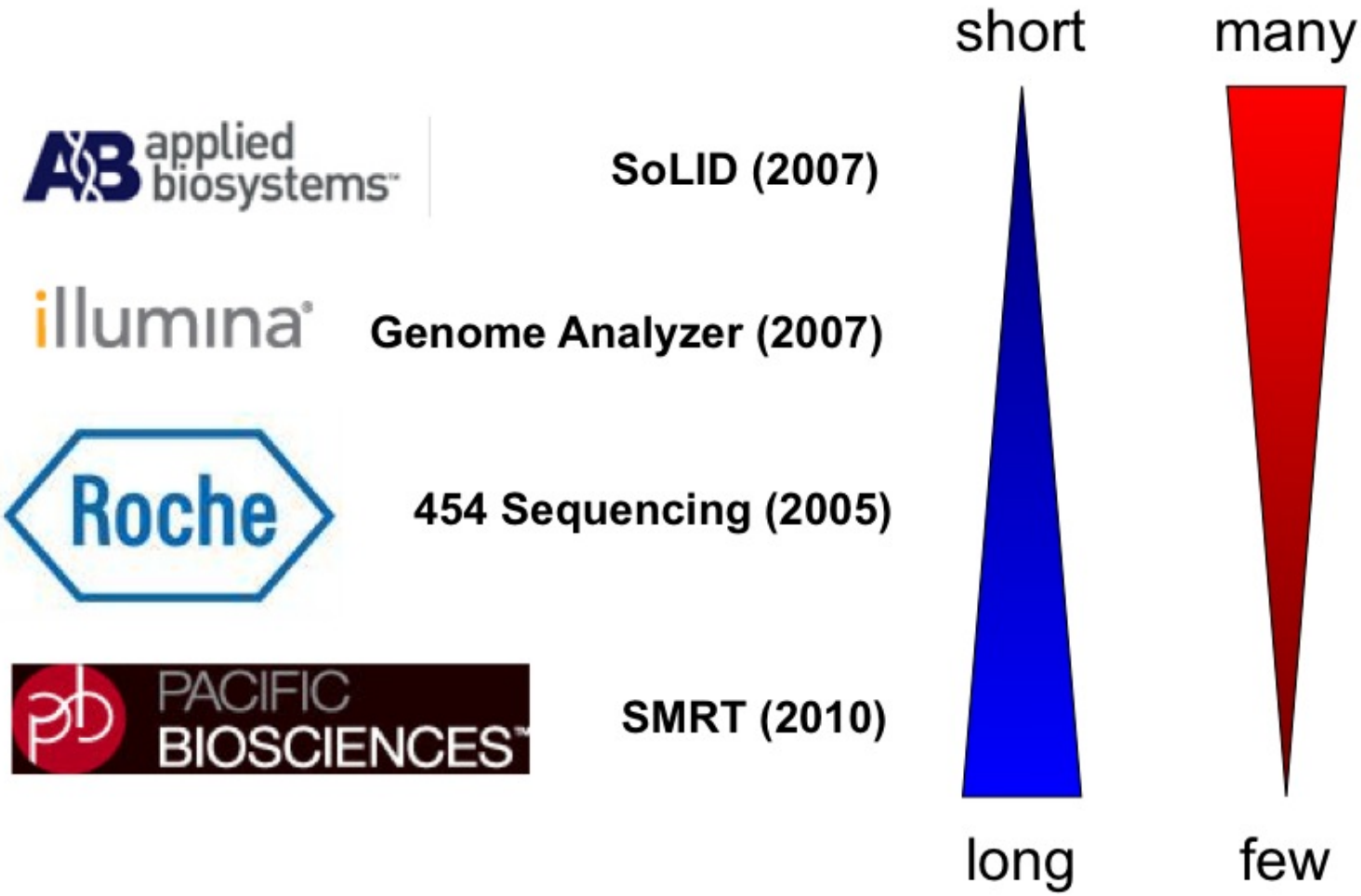
- Manual sequencing using gels was quickly automated by the mid-1980s.
  - Applied Biosystems
  - Capillary sequencing
  - 150-200 bp at first, paired 600-700 bp now
- A concerted effort from the NIH to sequence genomes of model organisms:
  - E coli (bacteria) 4.5 Mb (1997)
  - S cerevisiae (yeast) 6.0 Mb (1997)
  - C elegans (nematode worm) 98 Mb (1998)
  - Human 3 Gb (2000)
    - Estimated cost: $2.7 billion in 1991 dollars
    - Estimated time in 1990: 15 years

An Investment: the $1000 human genome

- The situation by the turn of the century:
  - Cost of a human-size draft genome (8x) in 2003: $50M
  - 4 main publicly supported genome centers in the US received the bulk of the money set aside for sequencing:
    - MIT (Broad)
    - Washington University
    - Baylor
    - DOE
- In 2003, NHGRI committed to develop next-generation sequencing technologies to lower the cost of 30x a human genome (~100 Gbp):
  - $100,000 genome
  - $1,000 genome
- Originally targeted for *de novo* sequencing, and resequencing for population genetics.

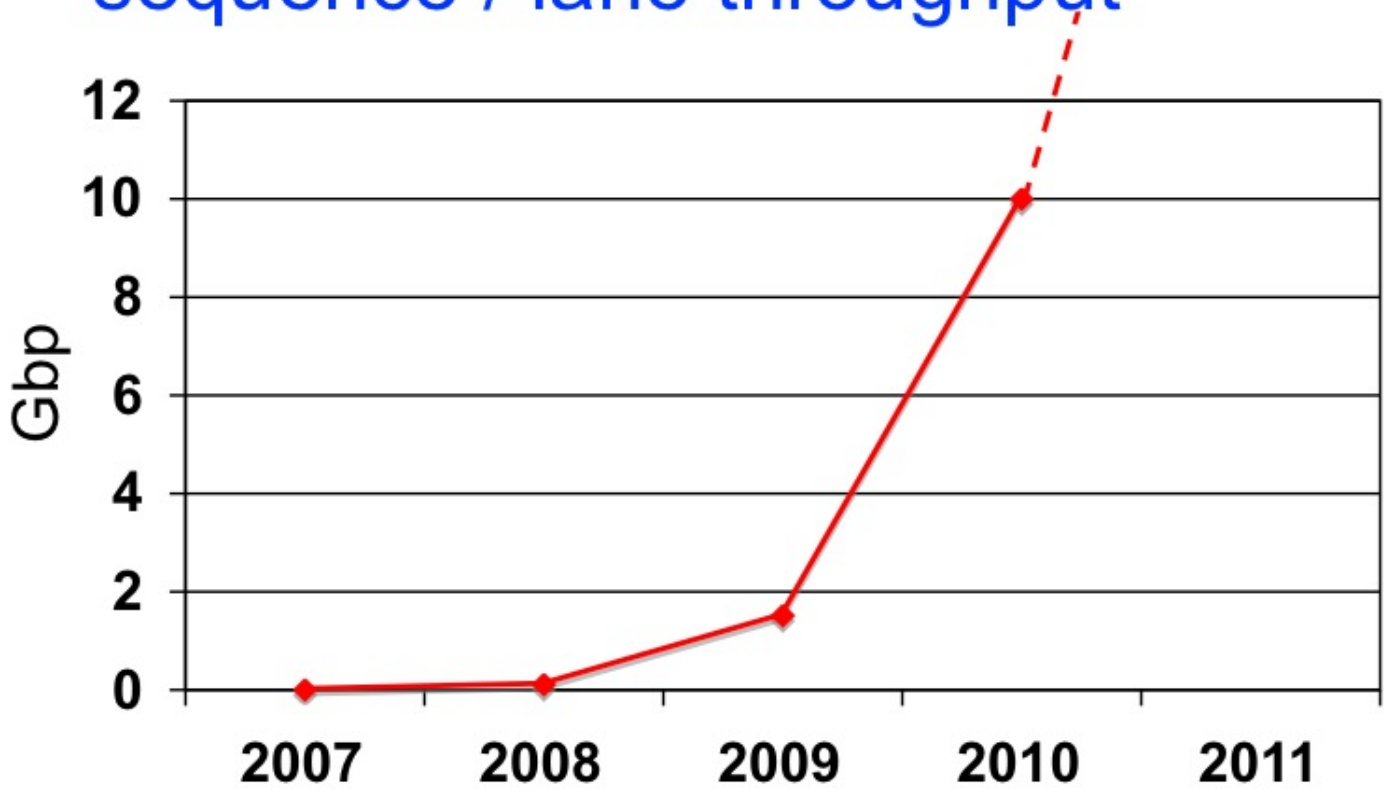


A variety of current technologies are available with different tradeoffs



Exponential growth of Illumina mapped sequence / lane throughput





	Jan-07	Jan-08	Jan-09	Jan-10	Jan-11
Gbp / lane	0.025	0.128	1.5	10	30-50 ?
Read type:	1x25	1x36	2x75	2x100	2x150

Cost/lane is relatively stable at $600 to $1,200

Genomes Today

- Sequencing (in small pieces) became very cheap
  - not very far from the $1000 human genome
  - <$1000 (small) genomes, e.g., worm, fly
  - High data storage & computing demands
- Many of genomes are sequenced per year
  - 1000 human genomes, 1001 Arabidopsis genomes
  - Hundreds of new species' genomes
- Challenges
  - Assembly & variant discovery
  - Association with phenotypes
  - Genome (re)annotation (e.g., identification of genes)
  - Functional annotation of genes or genome regions
  - Integration of heterogeneous data
  - Visualization beyond the single genome

Genomes Today

- Sequencing (in small pieces) became very cheap
  - not very far from the $1000 human genome
  - <$1000 (small) genomes, e.g., worm, fly
  - High data storage & computing demands
- Many of genomes are sequenced per year
  - 1000 human genomes, 1001 Arabidopsis genomes
  - Hundreds of new species' genomes
- Challenges
  - Assembly & variant discovery
  - Association with phenotypes
  - *Genome (re-)annotation (e.g., identification of genes)*
  - *Functional annotation of genes or genomic regions*
  - *Integration of heterogeneous data*
  - *Visualization beyond the third genome*

Genomes Today



- Sequencing (in small pieces) became very cheap
  - not very far from the $1000 human genome
  - <$1000 (small) genomes, e.g., worm, fly
  - High data storage & computing demands
- Many of genomes are sequenced per year
  - 1000 human genomes, 1001 Arabidopsis genomes
  - Hundreds of new species' genomes
- Challenges
  - Assembly & variant discovery
  - Association with phenotypes
  - *Genome (re-)annotation (e.g., identification of genes)*
  - *Functional annotation of genes or genomic regions*
  - *Integration of heterogeneous data*
  - Visualization beyond the single genome

Genomes Today



- Sequencing (in small pieces) became very cheap
  - not very far from the $1000 human genome
  - <$1000 (small) genomes, e.g., worm, fly
  - High data storage & computing demands
- Many of genomes are sequenced per year
  - 1000 human genomes, 1001 Arabidopsis genomes
  - Hundreds of new species' genomes
- Challenges
  - Assembly & variant discovery
  - Association with phenotypes
  - *Genome (re-)annotation (e.g., identification of genes)*
  - *Functional annotation of genes or genomic regions*
  - *Integration of heterogeneous data*
  - Visualization beyond the single genome

Genomes Today



- Sequencing (in small pieces) became very cheap
  - not very far from the $1000 human genome
  - <$1000 (small) genomes, e.g., worm, fly
  - High data storage & computing demands
- Many of genomes are sequenced per year
  - 1000 human genomes, 1001 Arabidopsis genomes
  - Hundreds of new species' genomes
- Challenges
  - Assembly & variant discovery
  - Association with phenotypes
  - *Genome (re-)annotation (e.g., identification of genes)*
  - *Functional annotation of genes or genomic regions*
  - *Integration of heterogeneous data*
  - Visualization beyond the single genome

Genomes Today



- Sequencing (in small pieces) became very cheap
  - not very far from the $1000 human genome
  - <$1000 (small) genomes, e.g., worm, fly
  - High data storage & computing demands
- Many of genomes are sequenced per year
  - 1000 human genomes, 1001 Arabidopsis genomes
  - Hundreds of new species' genomes
- Challenges
  - Assembly & variant discovery
  - Association with phenotypes
  - *Genome (re-)annotation (e.g., identification of genes)*
  - *Functional annotation of genes or genomic regions*
  - *Integration of heterogeneous data*
  - Visualization beyond the single genome

Read Assembly



Problem

Given a set of genomic reads, find maximally overlapping parts in order to reconstruct the original sequences

Classical assembly ⇒ **Too inefficient for NGS reads**

1. *Overlap phase*: Every read is compared with every other read and the overlap graph is computed
2. *Layout phase*: Pairs are determined that position every read in the assembly
3. *Consensus phase*: Multi-alignment of all the placed reads is produced to obtain the final sequence

New techniques based on **de Bruijn Graphs**

- Genome assembly: EULER [Pevzner et al., 2001], Velvet [Zerbino and Birney, 2008], Allpaths [Maccallum et al., 2009], SHARCGS [Dohm et al., 2007], SSAKE/VCAKE [Warren et al., 2007], ...
- Transcriptome assembly: Oases, Ananas

Results for Genome Assembly

Table 3. Comparison of short read assemblers on experimental *Streptococcus suis* Solexa reads

Assembler	No. of contigs	N50	Average error rate	Memory	Time	Seq. Cov.
Velvet 0.3	470	8661 bp	0.02%	2.0G	2 min 57 sec	97%
SSAKE 2.0	265	1727 bp	0.20%	1.7G	1 h 47 min	16%
VCAKE 1.0	7675	1137 bp	0.64%	1.8G	4 h 25 min	134%

Considerably shorter fragments for larger genomes

Considerably longer fragments for reference-guided assemblies

[Zerbino and Birney, 2008]

Results for Genome Assembly



Table 3. Comparison of short read assemblers on experimental *Streptococcus suis* Solexa reads

Assembler	No. of contigs	N50	Average error rate	Memory	Time	Seq. Cov.
Velvet 0.3	470	8661 bp	0.02%	2.0G	2 min 57 sec	97%
SSAKE 2.0	265	1727 bp	0.20%	1.7G	1 h 47 min	16%
VCAKE 1.0	7675	1137 bp	0.64%	1.8G	4 h 25 min	134%

Considerably shorter fragments for larger genomes

Considerably longer fragments for reference-guided assemblies

[Zerbino and Birney, 2008]

Results for Genome Assembly

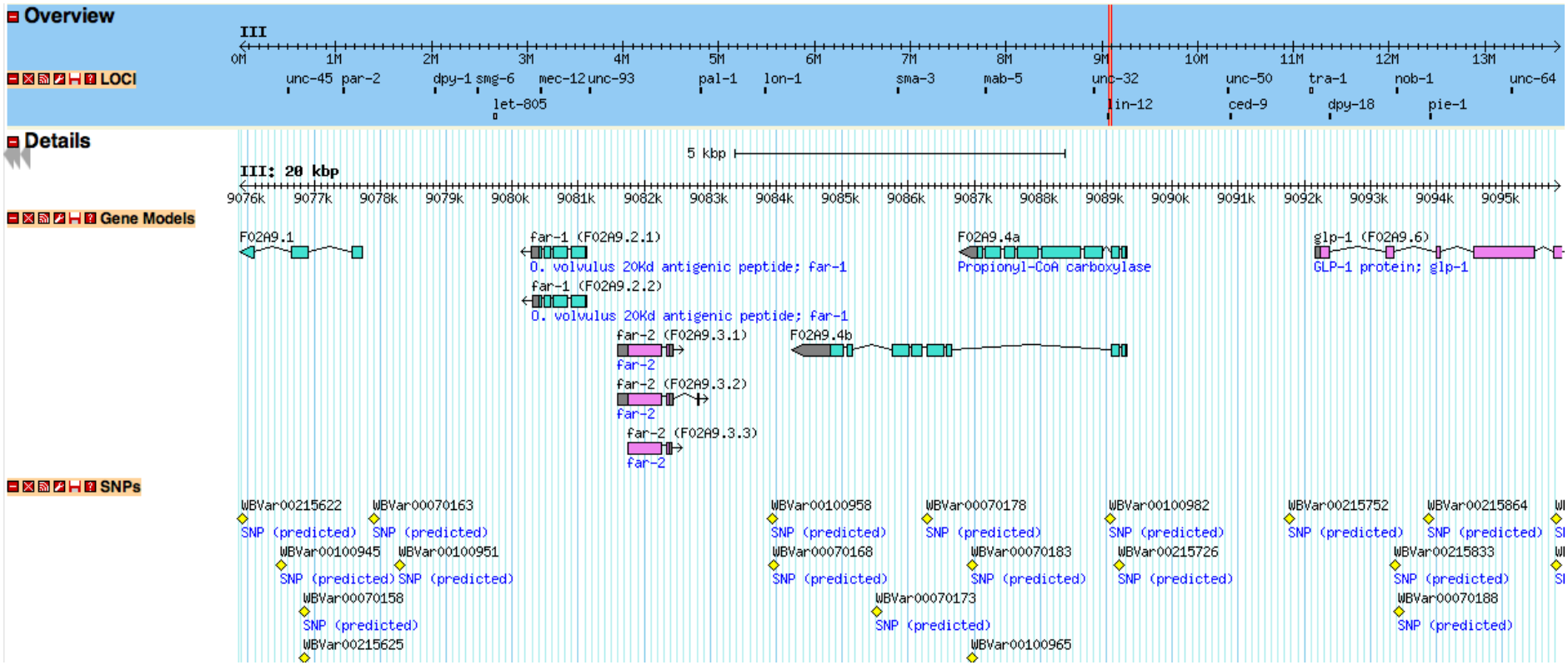
Table 3. Comparison of short read assemblers on experimental *Streptococcus suis* Solexa reads

Assembler	No. of contigs	N50	Average error rate	Memory	Time	Seq. Cov.
Velvet 0.3	470	8661 bp	0.02%	2.0G	2 min 57 sec	97%
SSAKE 2.0	265	1727 bp	0.20%	1.7G	1 h 47 min	16%
VCAKE 1.0	7675	1137 bp	0.64%	1.8G	4 h 25 min	134%

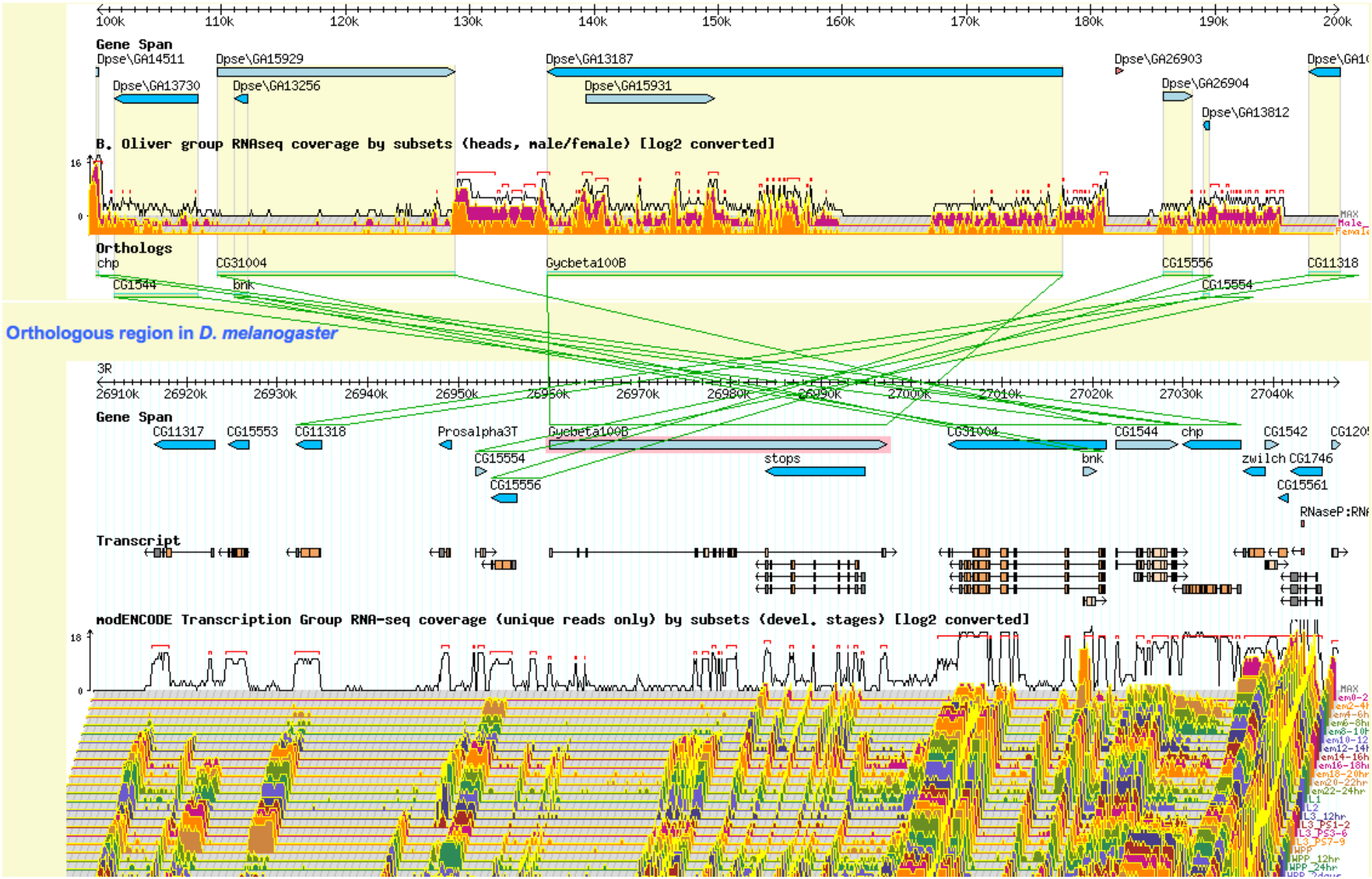
Considerably shorter fragments for larger genomes
Considerably longer fragments for reference-guided assemblies

[Zerbino and Birney, 2008]

Non-linear Genomes



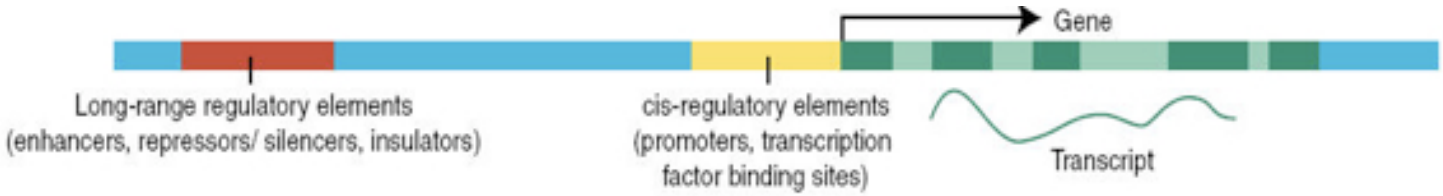
Non-linear Genomes



The Encyclopedia of DNA Elements



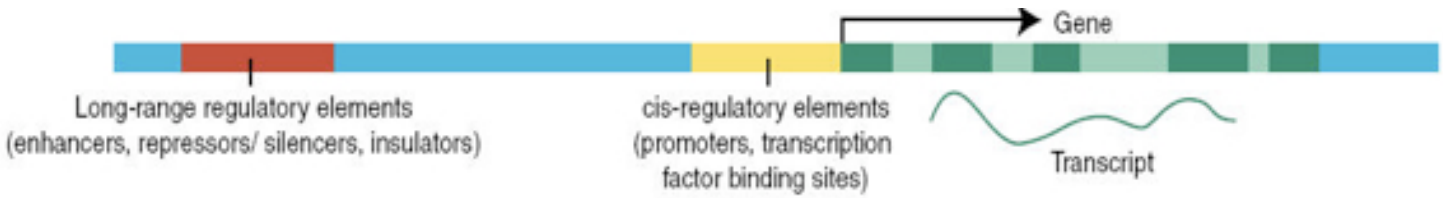
- How to make sense of the genome?
  - Which nucleotides are functional?
  - What is their function?



- National Human Genome Research Institute started the Encode project to annotate human and model organism genomes:
  - 2004: Human 1%
  - 2007: Human whole-genome
  - 2008: *Drosophila* and *C. elegans*
  - 2010: Mouse

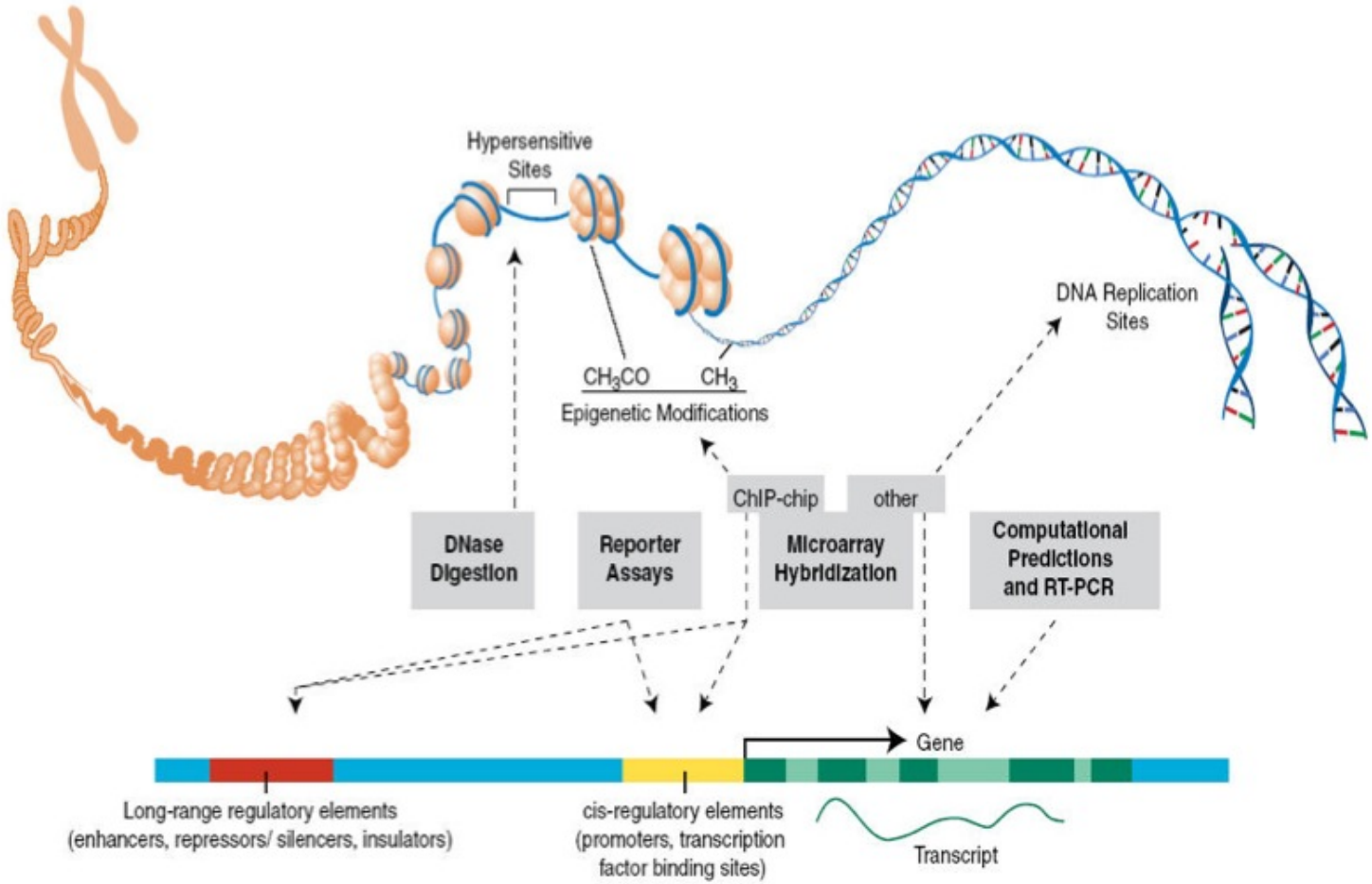
The Encyclopedia of DNA Elements

- How to make sense of the genome?
  - Which nucleotides are functional?
  - What is their function?



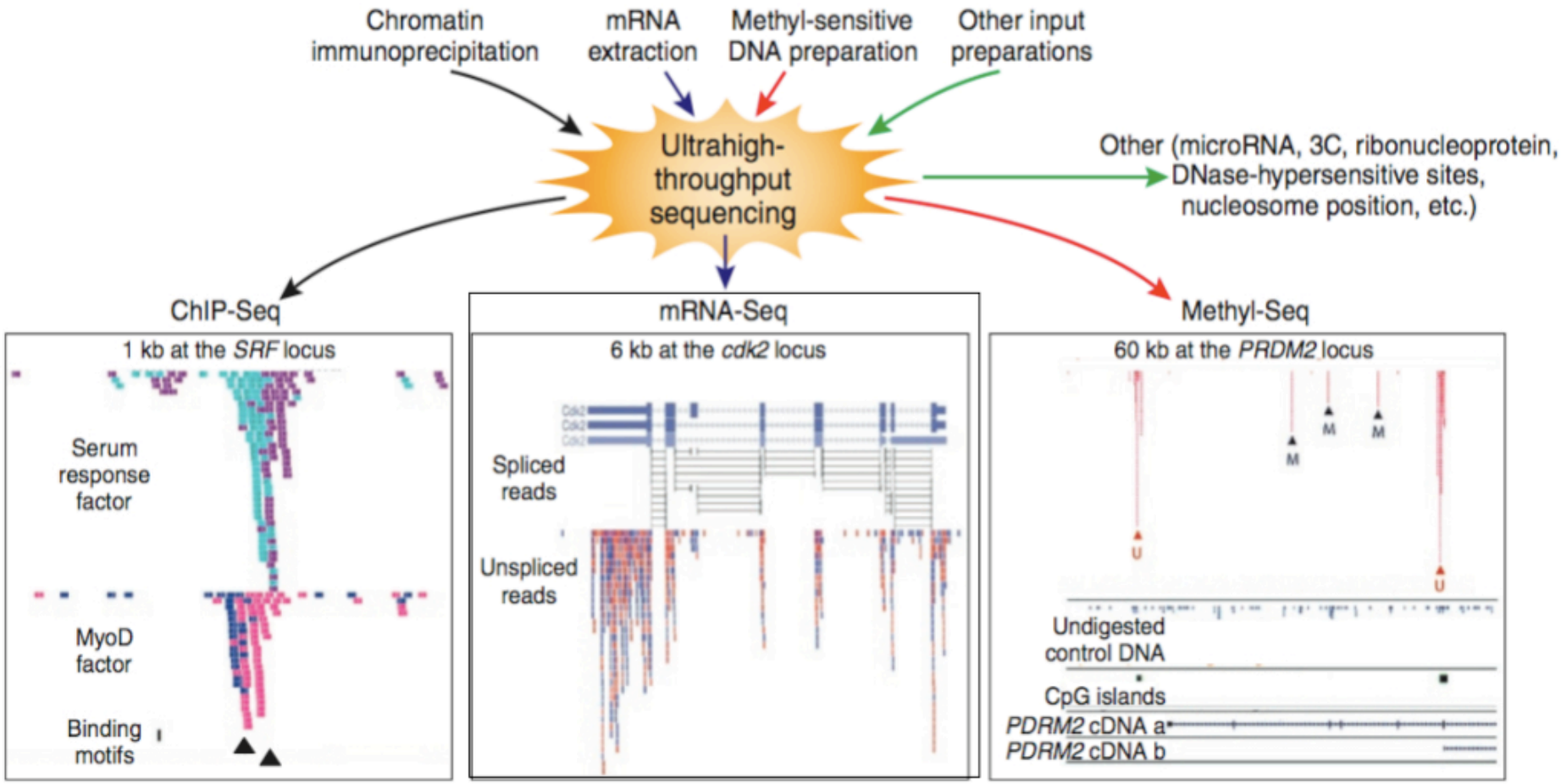
- National Human Genome Research Institute started the Encode project to annotate human and model organism genomes:
  - 2004: Human 1%
  - 2007: Human whole-genome
  - 2008: *Drosophila* and *C. elegans*
  - 2010: Mouse

Originally, microarrays were used to read out genome-wide functional assays



ENCODE Project Consortium (2004). Science 306: 636.

Sequencing for Functional Genomics



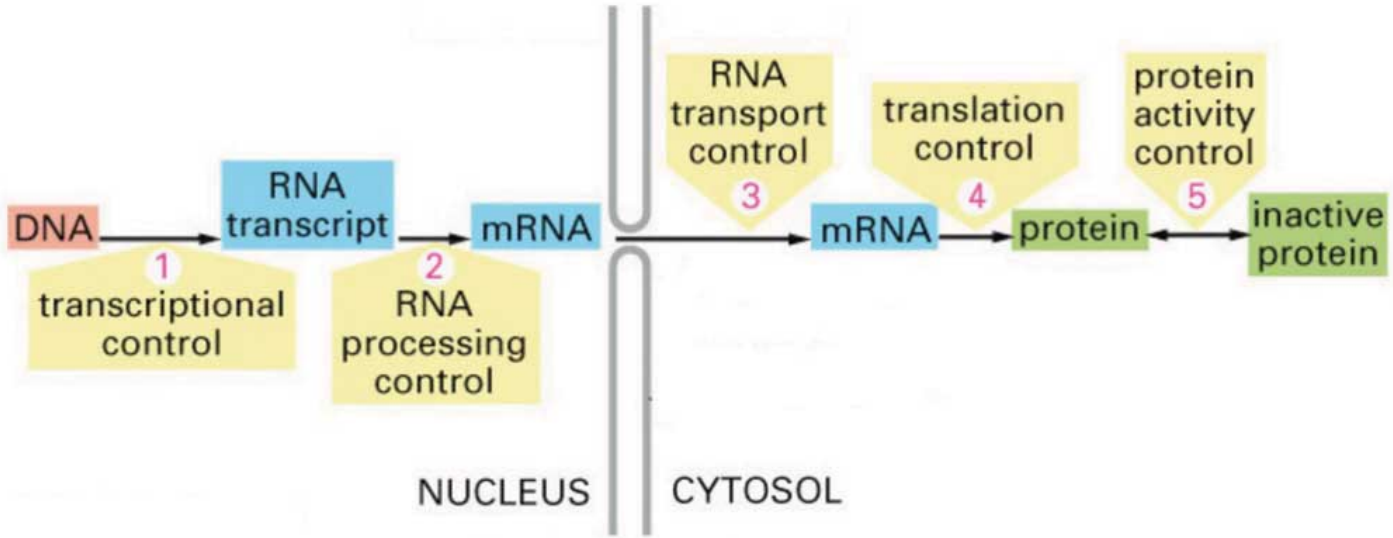
[Wold & Myers, Nature Methods, 2007]

Key Research Questions



- Characterize an organism's full complement of genes
  - $\Rightarrow$ Find new (possibly noncoding) genes
  - $\Rightarrow$ Compare genes among organisms
- Characterize transcript isoforms
  - $\Rightarrow$ Find new alternative splice forms / transcript ends
- Monitor transcriptome changes between tissues or in response to environmental changes (e.g. stress)
  - $\Rightarrow$ Identify significant expression changes
- Understand transcriptome regulation
  - $\Rightarrow$ Knock-out / knock-down analysis of regulators
    - $\rightsquigarrow$ Identify regulated targets with significant expression changes
  - $\Rightarrow$ Identify binding sites used in regulation (e.g. ChIP-seq)

Learning about the Transcriptome



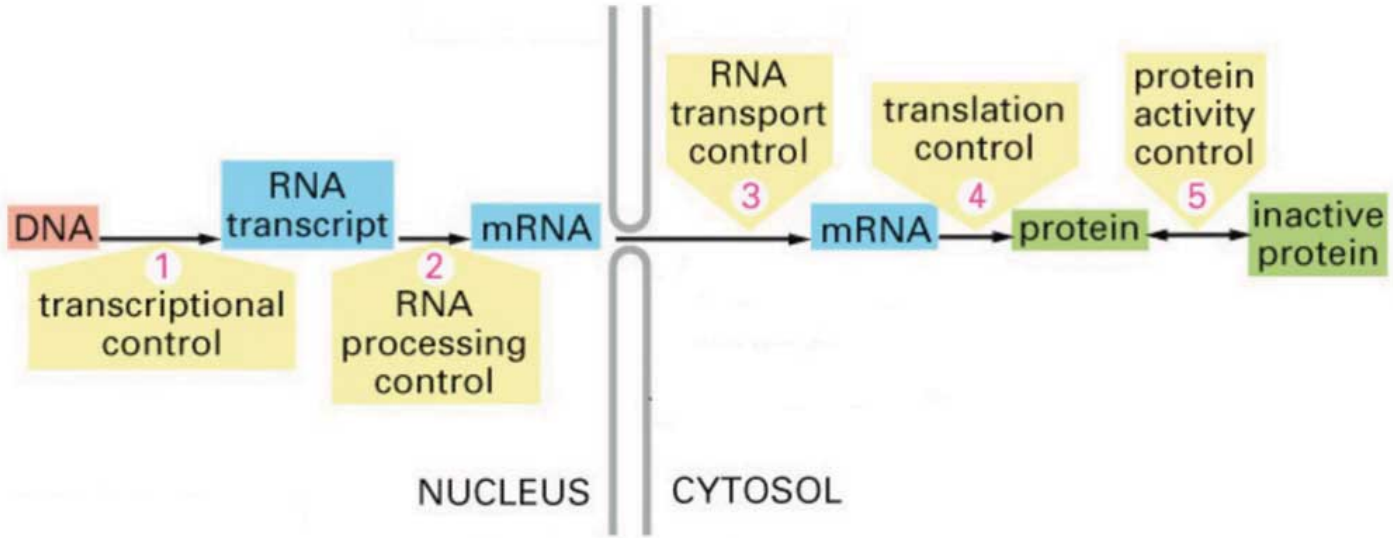
Computational Point of View

- How to learn to predict what these processes accomplish?
- How well can we predict it from the available information?

Biological View

- What can we not predict yet? What is missing?
- Can we derive a deeper understanding of these processes?

Learning about the Transcriptome



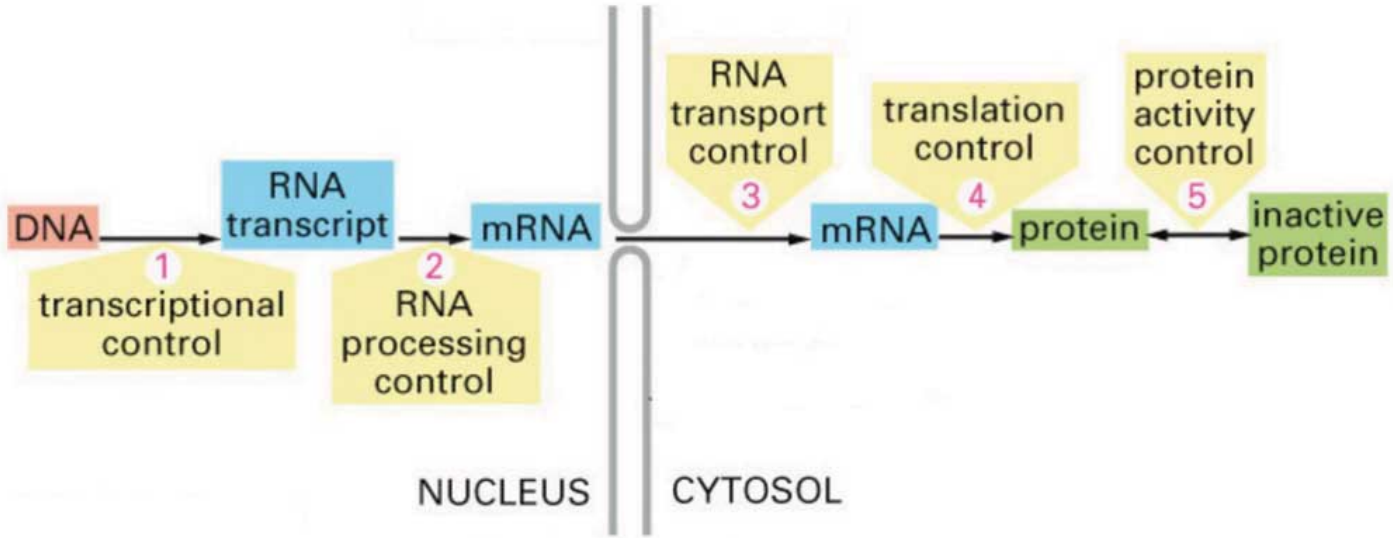
Computational Point of View

- How to learn to predict what these processes accomplish?
- How well can we predict it from the available information?

Biological View

- What can we not predict yet? What is missing?
- Can we derive a deeper understanding of these processes?

Learning about the Transcriptome



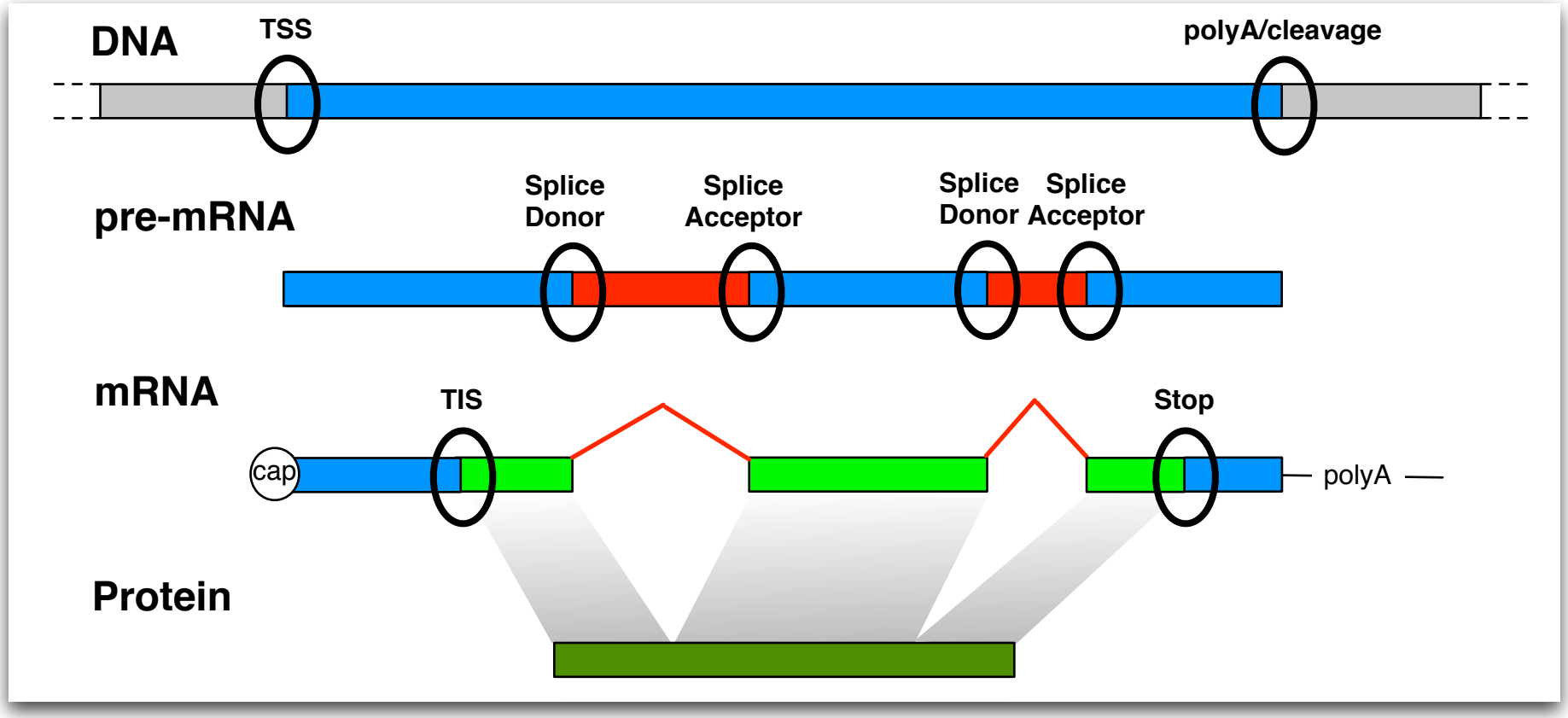
Computational Point of View

- How to learn to predict what these processes accomplish?
- How well can we predict it from the available information?

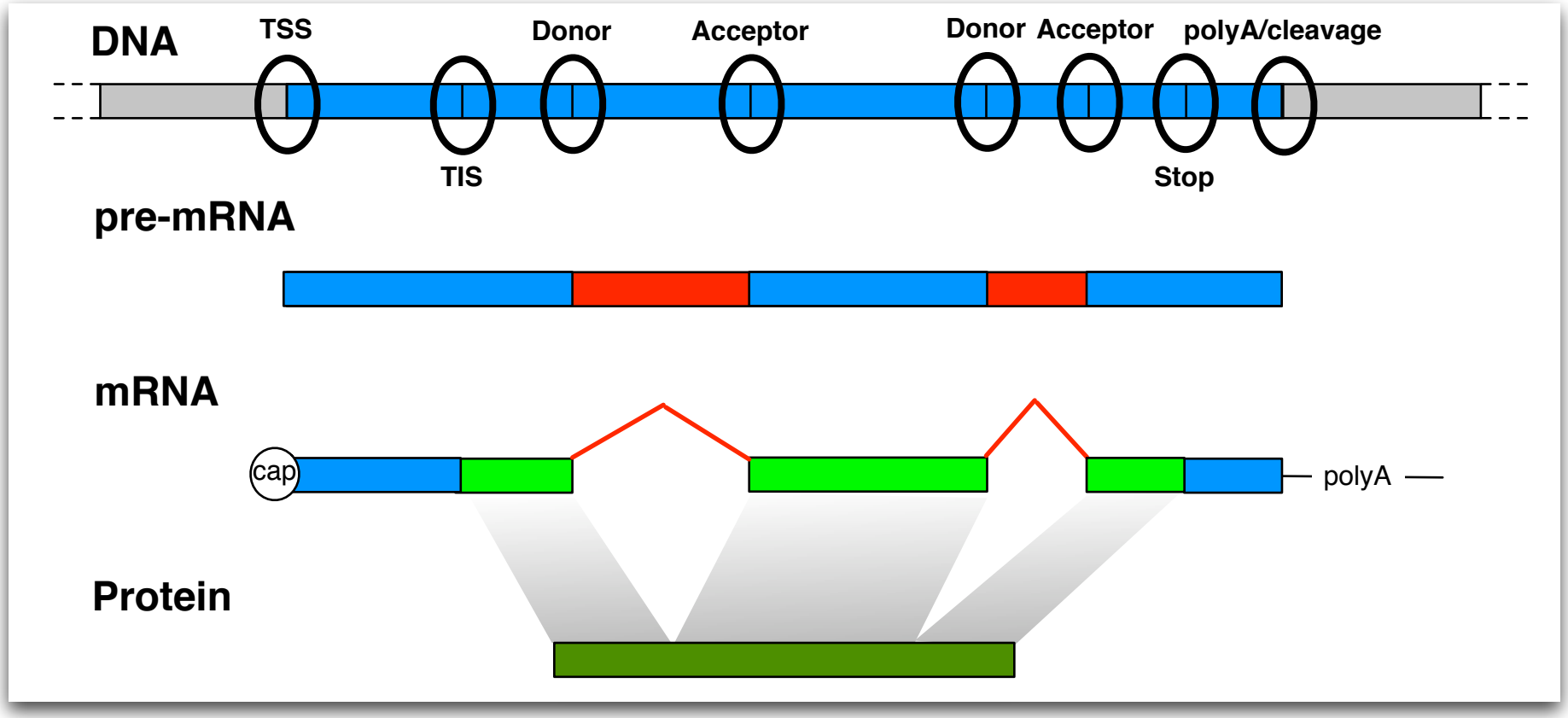
Biological View

- What can we not predict yet? What is missing?
- Can we derive a deeper understanding of these processes?

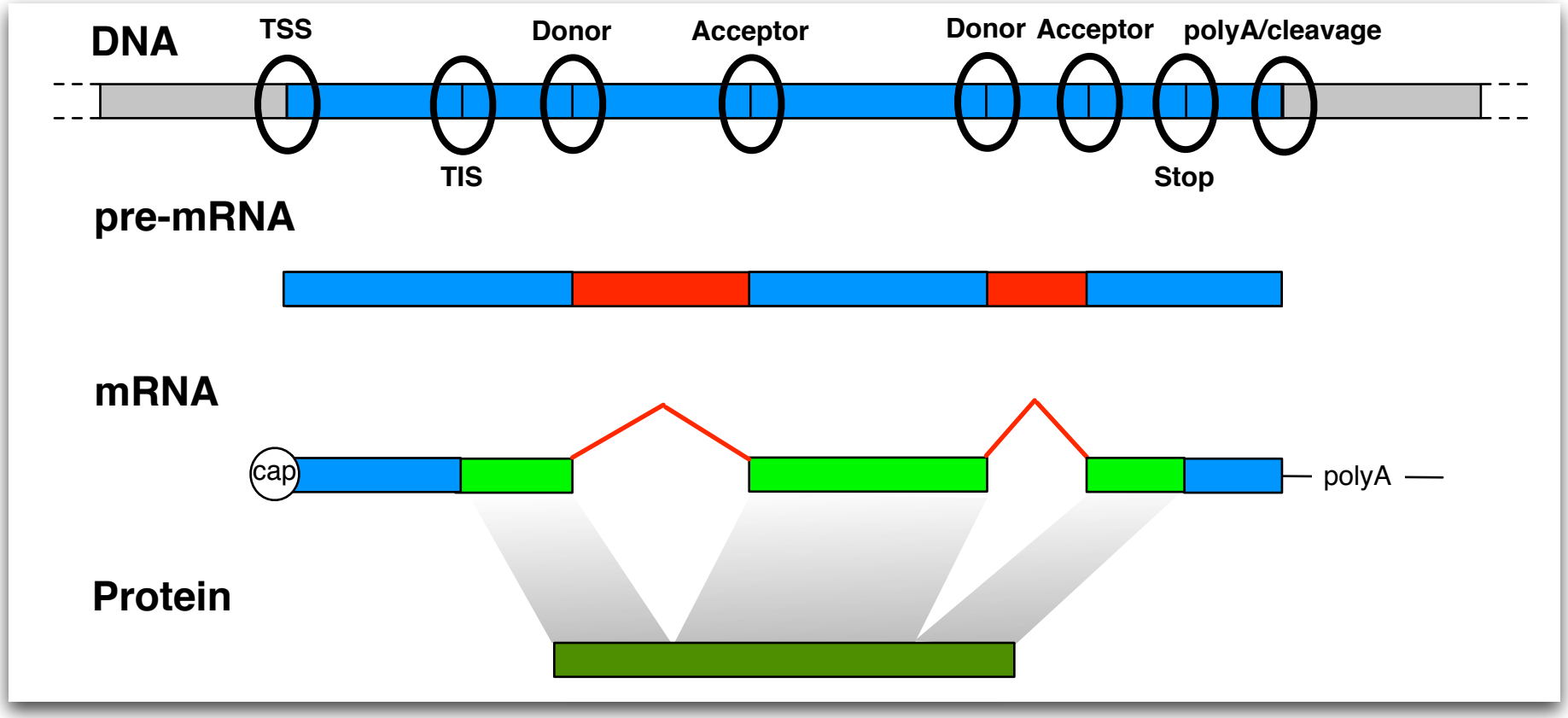
1st Step: Computational Gene Prediction

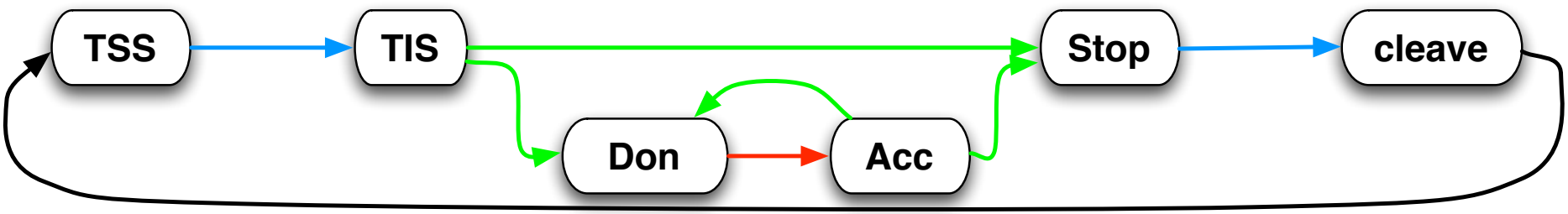


1st Step: Computational Gene Prediction



1st Step: Computational Gene Prediction





What do we need?



Predictions:

- Signal site recognition (transcription starts, splice sites, etc.)
- Characterization of sequence regions
- Transcripts/Gene structures
- Models of regulation (transcriptional, splicing etc.)

Tools:

- Sequence classification
  - SVMs with appropriate string kernels
- Sequence segmentation
  - Discriminative learning to predict structured outputs

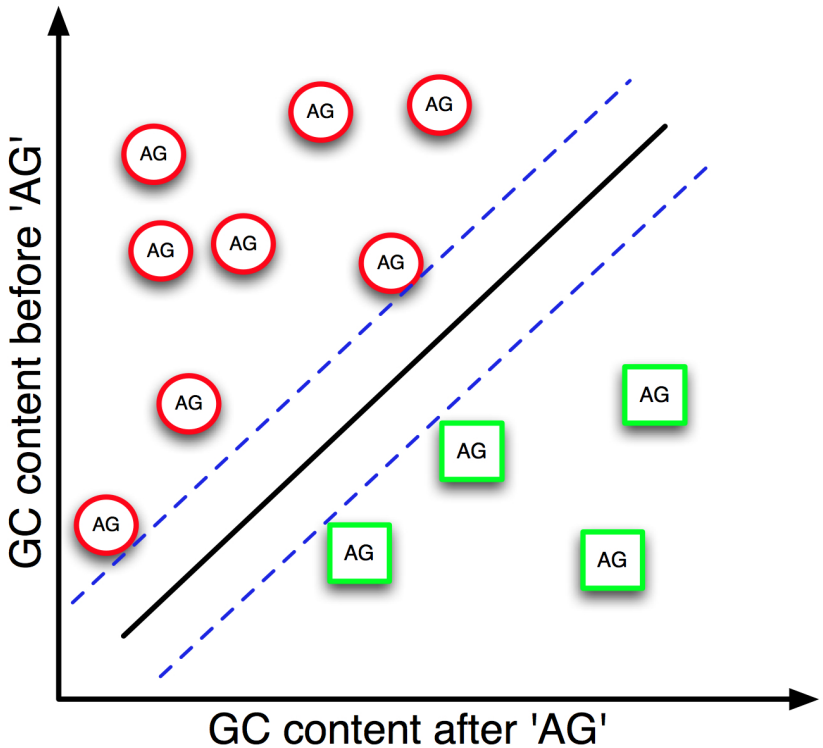
Example: Recognition of Splice Sites

- Given: Potential acceptor splice sites

```
AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC
```

intron **exon**

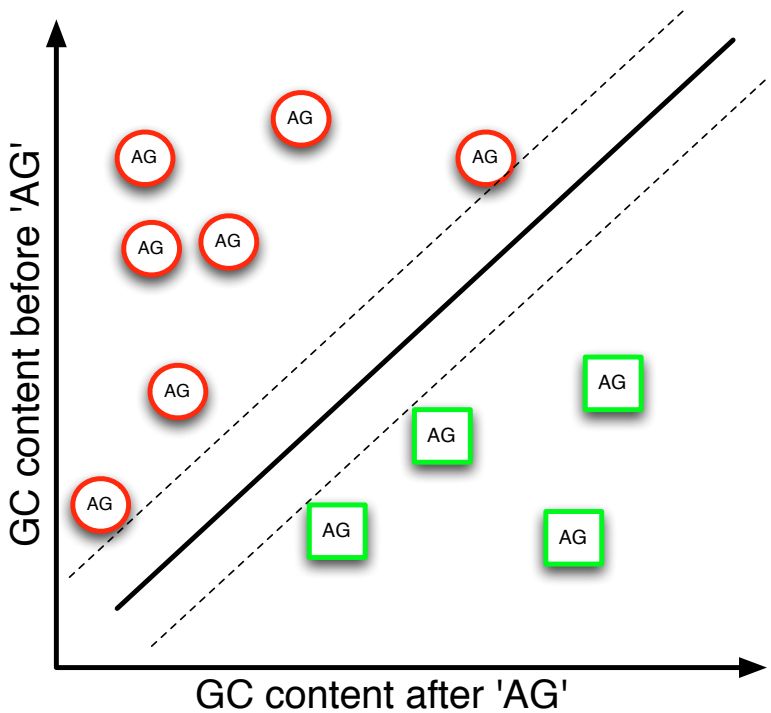
- Goal: Rule that distinguishes true from false ones



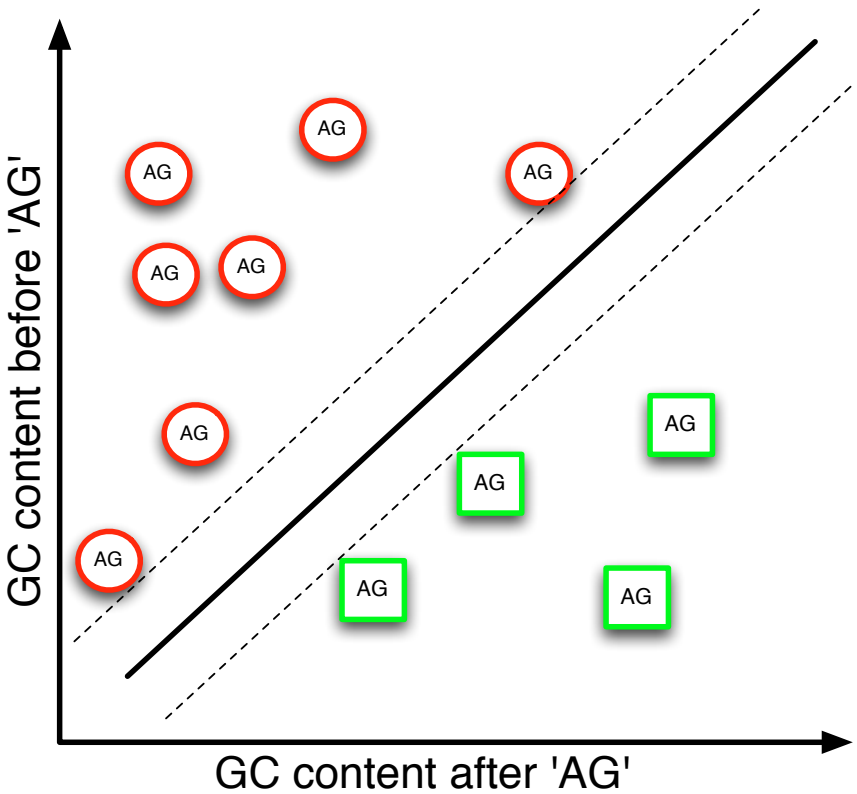
Linear Classifiers
with large **margin**

SVMs: Maximize the margin! Why?

- Intuitively, it feels the safest.
- For a small error in the separating hyperplane, we do not suffer too many mistakes.
- Empirically, it works well.
- Learning theory indicates that it is the right thing to do.



How to Maximize the Margin? I



Minimize

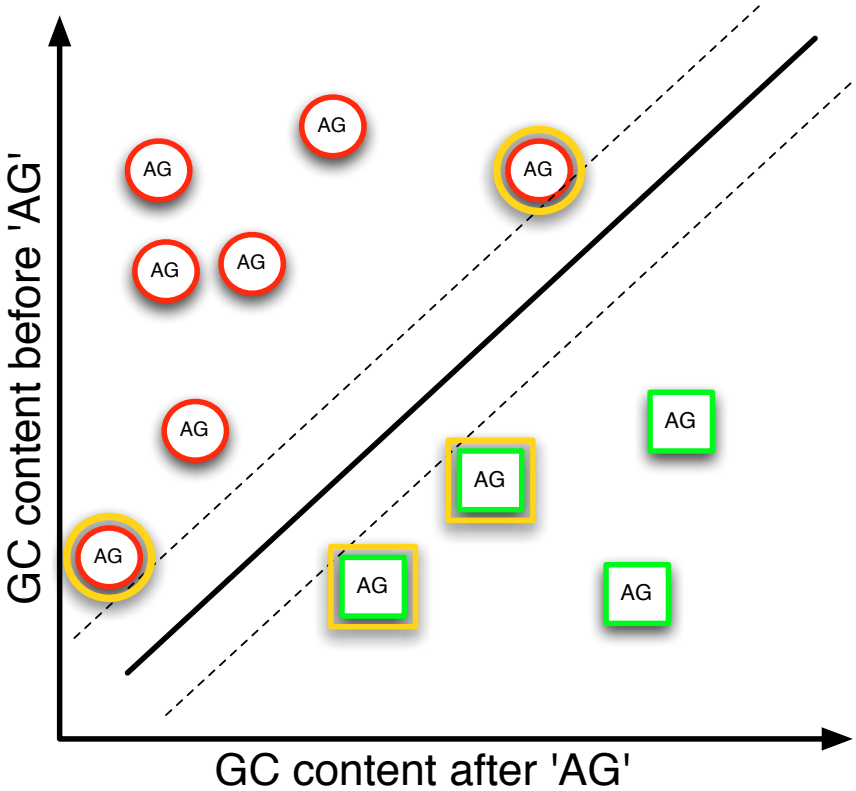
$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

Subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
for all $i = 1, \ldots, n$.

- Examples on the margin are called support vectors [Vapnik, 1995]
- Soft margin SVMs [Cortes and Vapnik, 1995]

How to Maximize the Margin? I



Minimize

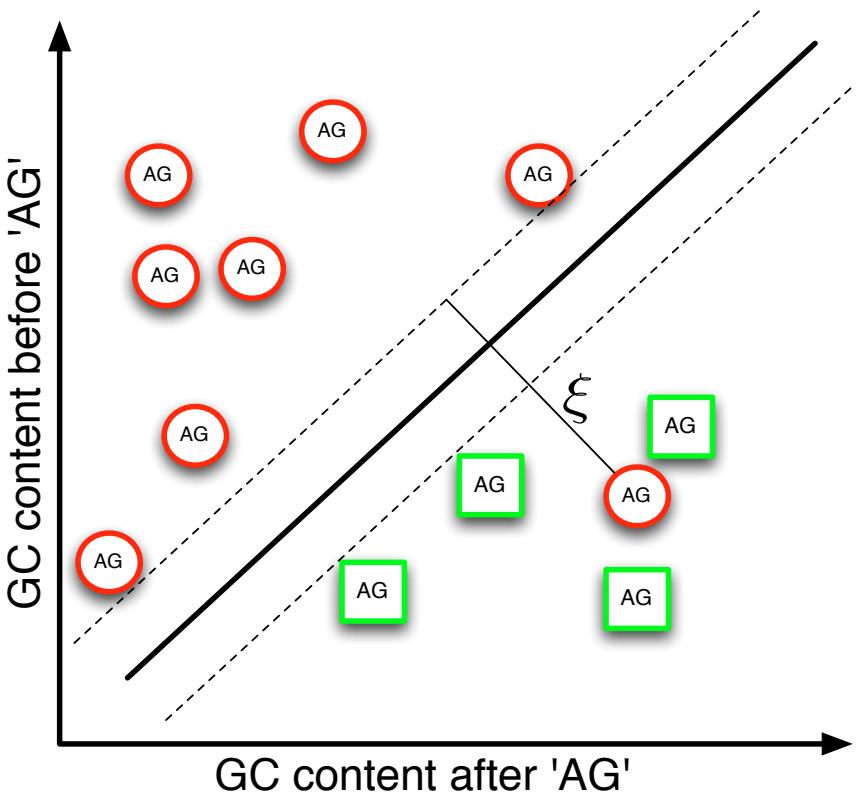
$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

Subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
for all $i = 1, \ldots, n$.

- Examples on the margin are called **support vectors** [Vapnik, 1995]
- Soft margin SVMs [Cortes and Vapnik, 1995]

How to Maximize the Margin? I



Minimize

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

Subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
for all $i = 1, \ldots, n$.

- Examples on the margin are called support vectors [Vapnik, 1995]
- Soft margin SVMs [Cortes and Vapnik, 1995]

An Important Detail

$$\underset{\mathbf{w},b,\boldsymbol{\xi}}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i \text{ for all } i = 1,\ldots,n.$$
$$\xi_i \geqslant 0 \text{ for all } i = 1,\ldots,n$$

Theorem: The optimal $\mathbf{w}$ can be written as a linear combination of the examples (for appropriate α's):

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i\mathbf{x}_i \qquad \Rightarrow \text{Plug in!}$$

Now optimize for the variables α, b, and ξ!

Corollary: Hyperplane only depends on the scalar products of the examples

$$\langle \mathbf{x}, \hat{\mathbf{x}}\rangle = \sum_{d=1}^{D} x_d\hat{x}_d \qquad \text{Recognition rule}$$

An Important Detail

$$\underset{\mathbf{w},b,\boldsymbol{\xi}}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i \text{ for all } i = 1,\ldots,n.$$
$$\xi_i \geqslant 0 \text{ for all } i = 1,\ldots,n$$

Theorem: The optimal $\mathbf{w}$ can be written as a linear combination of the examples (for appropriate α's):

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i\mathbf{x}_i \qquad \Rightarrow \text{Plug in!}$$

Now optimize for the variables α, b, and ξ!

Corollary: Hyperplane only depends on the scalar products of the examples

$$\langle \mathbf{x}, \hat{\mathbf{x}}\rangle = \sum_{d=1}^{D} x_d\hat{x}_d \qquad \text{Recognition rule}$$

An Important Detail

$$\underset{\boldsymbol{\alpha},b,\boldsymbol{\xi}}{\text{minimize}} \quad \frac{1}{2}\left\|\sum_{i=1}^{N}\alpha_i\mathbf{x}_i\right\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \quad y_i\left(\sum_{j=1}^{N}\alpha_j\langle\mathbf{x}_j,\mathbf{x}_i\rangle + b\right) \geqslant 1-\xi_i \text{ for all } i=1,\ldots,n.$$
$$\xi_i \geqslant 0 \text{ for all } i=1,\ldots,n$$

Theorem: The optimal $\mathbf{w}$ can be written as a linear combination of the examples (for appropriate α's):

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i\mathbf{x}_i \qquad \Rightarrow \text{Plug in!}$$

Now optimize for the variables $\boldsymbol{\alpha}$, b, and $\boldsymbol{\xi}$!

Corollary: Hyperplane only depends on the scalar products of the examples

$$\langle\mathbf{x},\hat{\mathbf{x}}\rangle = \sum_{d=1}^{D}x_d\hat{x}_d \qquad \text{Remember this!}$$

An Important Detail

$$\underset{\boldsymbol{\alpha},b,\boldsymbol{\xi}}{\text{minimize}} \quad \frac{1}{2}\left\|\sum_{i=1}^{N}\alpha_i\mathbf{x}_i\right\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \quad y_i\left(\sum_{j=1}^{N}\alpha_j\langle\mathbf{x}_j,\mathbf{x}_i\rangle + b\right) \geqslant 1-\xi_i \text{ for all } i=1,\ldots,n.$$
$$\xi_i \geqslant 0 \text{ for all } i=1,\ldots,n$$

Theorem: The optimal $\mathbf{w}$ can be written as a linear combination of the examples (for appropriate α's):

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i\mathbf{x}_i \qquad \Rightarrow \text{Plug in!}$$

Now optimize for the variables $\boldsymbol{\alpha}$, b, and $\boldsymbol{\xi}$!

Corollary: Hyperplane only depends on the scalar products of the examples

$$\langle\mathbf{x},\hat{\mathbf{x}}\rangle = \sum_{d=1}^{D}x_d\hat{x}_d$$

Remember this!

An Important Detail

$$\underset{\boldsymbol{\alpha},b,\boldsymbol{\xi}}{\text{minimize}} \quad \frac{1}{2}\left\|\sum_{i=1}^{N}\alpha_i\mathbf{x}_i\right\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \quad y_i\left(\sum_{j=1}^{N}\alpha_j\langle\mathbf{x}_j,\mathbf{x}_i\rangle + b\right) \geqslant 1-\xi_i \text{ for all } i=1,\ldots,n.$$
$$\xi_i \geqslant 0 \text{ for all } i=1,\ldots,n$$

Theorem: The optimal $\mathbf{w}$ can be written as a linear combination of the examples (for appropriate α's):

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i\mathbf{x}_i \qquad \Rightarrow \text{Plug in!}$$

Now optimize for the variables $\boldsymbol{\alpha}$, b, and $\boldsymbol{\xi}$!

Corollary: Hyperplane only depends on the scalar products of the examples

$$\langle\mathbf{x},\hat{\mathbf{x}}\rangle = \sum_{d=1}^{D}x_d\hat{x}_d \qquad \text{Remember this!}$$

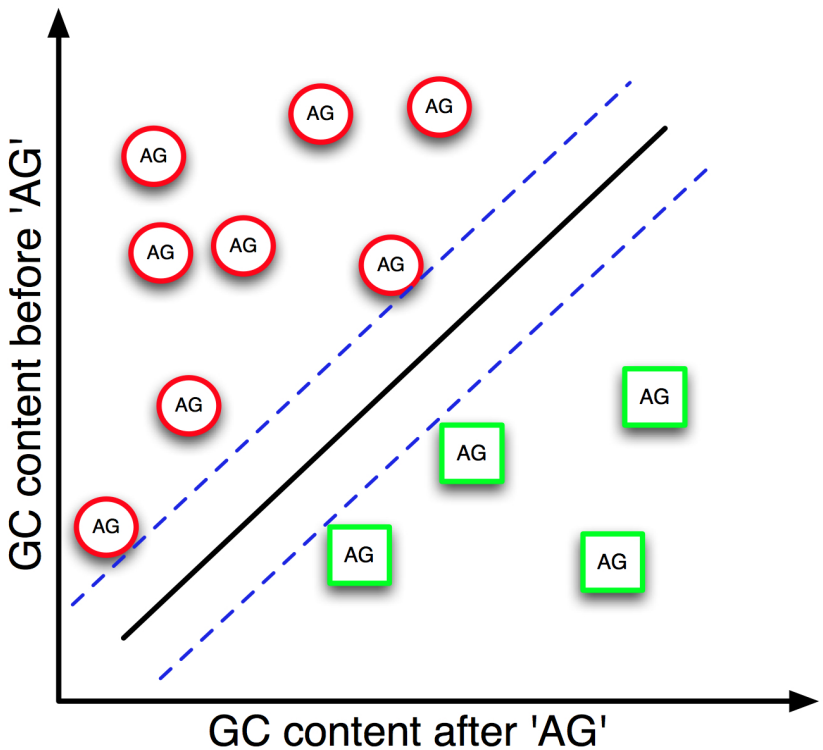
Recognition of Splice Sites

- Given: Potential acceptor splice sites

```
AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC
```

intron **exon**

- Goal: Rule that distinguishes true from false ones



Linear Classifiers
with large **margin**

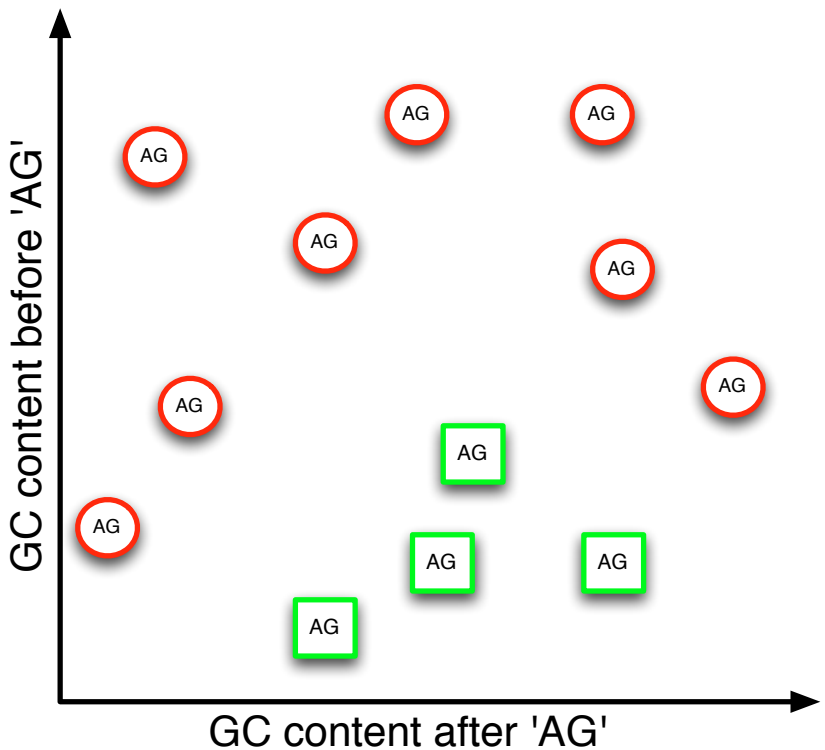
Recognition of Splice Sites

- Given: Potential acceptor splice sites

```
AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC
```

intron **exon**

- Goal: Rule that distinguishes true from false ones



More realistic problem?

- Not linearly separable!
- Need nonlinear separation?
- Need more features?

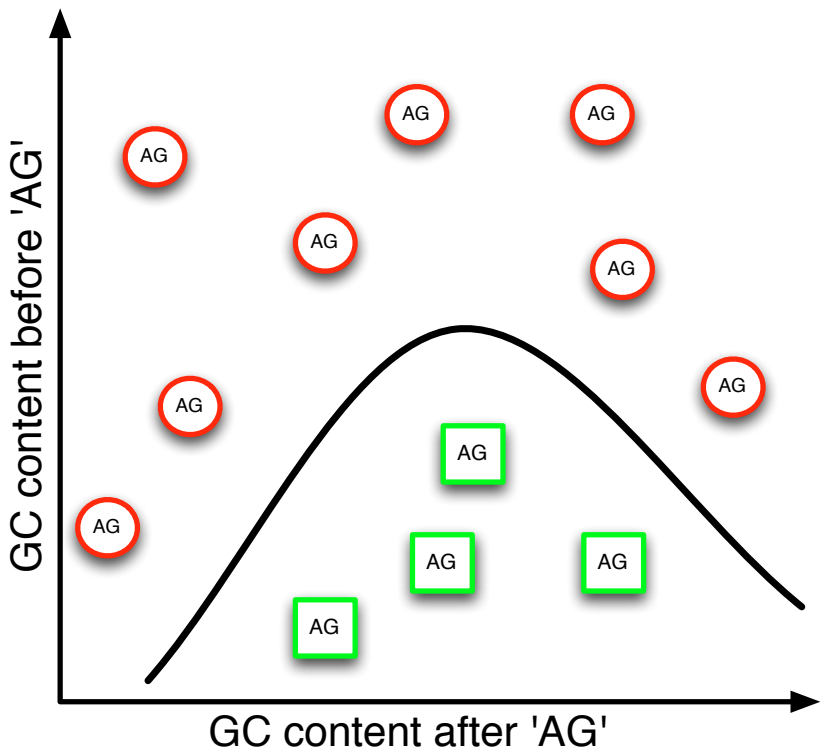
Recognition of Splice Sites

- Given: Potential acceptor splice sites

```
AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC
```

intron **exon**

- Goal: Rule that distinguishes true from false ones



More realistic problem?

- Not linearly separable!
- **Need nonlinear separation?**
- Need more features?

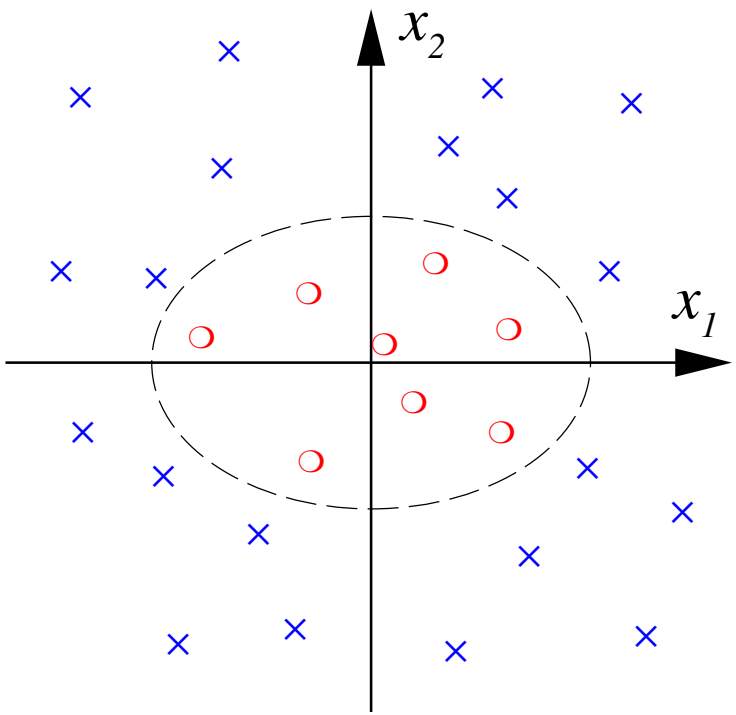
Nonlinear Separations

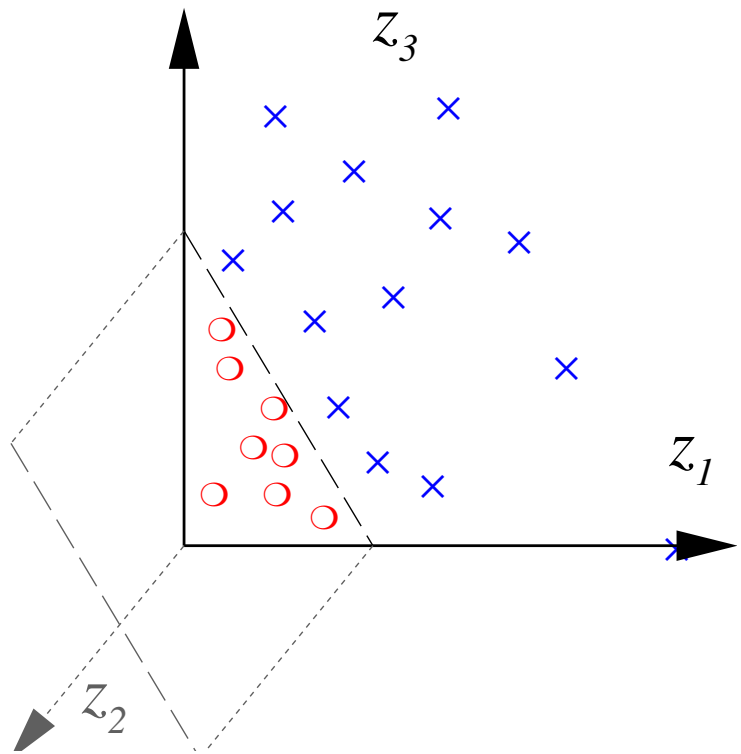
Linear separation might not be sufficient!

$\Rightarrow$ Map into a higher dimensional feature space

Example: all second order monomials

$$\begin{aligned} \Phi : \mathbb{R}^2 &\rightarrow \mathbb{R}^3 \\ (x_1, x_2) &\mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2) \end{aligned}$$





Kernel "Trick"



Example: $\mathbf{x} \in \mathbb{R}^2$ and $\Phi(\mathbf{x}) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$ [Boser et al., 1992]

$$\begin{aligned}
\langle \Phi(\mathbf{x}), \Phi(\hat{\mathbf{x}}) \rangle &= \left\langle (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2), (\hat{x}_1^2, \sqrt{2}\, \hat{x}_1 \hat{x}_2, \hat{x}_2^2) \right\rangle \\
&= \langle (x_1, x_2), (\hat{x}_1, \hat{x}_2) \rangle^2 \\
&= \langle \mathbf{x}, \hat{\mathbf{x}} \rangle^2 \\
&=: \mathrm{k}(\mathbf{x}, \hat{\mathbf{x}})
\end{aligned}$$

- Scalar product in feature space (here $\mathbb{R}^3$) can be computed in input space (here $\mathbb{R}^2$)!
- Also works for higher orders and dimensions
  $\Rightarrow$ relatively low-dimensional input spaces
  $\Rightarrow$ very high-dimensional feature spaces

Putting Things Together ...

- Use $\Phi(\mathbf{x})$ instead of $\mathbf{x}$
- Use linear classifier on the $\Phi(\mathbf{x})$'s
- From theorem: $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i)$.
- Nonlinear separation:

$$\begin{aligned} f(\mathbf{x}) &= \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b \\ &= \sum_{i=1}^{n} \alpha_i \underbrace{\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) \rangle}_{\mathrm{k}(\mathbf{x}_i, \mathbf{x})} + b \end{aligned}$$

- Trick: $\mathrm{k}(\mathbf{x}, \hat{\mathbf{x}}) = \langle \Phi(\mathbf{x}), \Phi(\hat{\mathbf{x}}) \rangle$, i.e. **do not use Φ, but k!**

See e.g. Vapnik [1995], Müller et al. [2001], Schölkopf and Smola [2002] for details.

Common Kernels



See e.g. Vapnik [1995], Müller et al. [2001], Schölkopf and Smola [2002]

$$
\begin{aligned}
\text{Polynomial} \quad & \mathrm{k}(\mathbf{x}, \hat{\mathbf{x}}) &&= (\langle \mathbf{x}, \hat{\mathbf{x}} \rangle + c)^d \\
\text{Sigmoid} \quad & \mathrm{k}(\mathbf{x}, \hat{\mathbf{x}}) &&= \tanh(\kappa \langle \mathbf{x}, \hat{\mathbf{x}} \rangle) + \theta) \\
\text{RBF} \quad & \mathrm{k}(\mathbf{x}, \hat{\mathbf{x}}) &&= \exp\left(-\|\mathbf{x} - \hat{\mathbf{x}}\|^2/(2\,\sigma^2)\right) \\
\text{Convex combinations} \quad & \mathrm{k}(\mathbf{x}, \hat{\mathbf{x}}) &&= \beta_1 \mathrm{k}_1(\mathbf{x}, \hat{\mathbf{x}}) + \beta_2 \mathrm{k}_2(\mathbf{x}, \hat{\mathbf{x}}) \\
\text{Normalization} \quad & \mathrm{k}(\mathbf{x}, \hat{\mathbf{x}}) &&= \frac{\mathrm{k}'(\mathbf{x}, \hat{\mathbf{x}})}{\sqrt{\mathrm{k}'(\mathbf{x}, \mathbf{x}) \mathrm{k}'(\hat{\mathbf{x}}, \hat{\mathbf{x}})}}
\end{aligned}
$$

Notes:

- Kernels may be combined in case of heterogeneous data
- These kernels are good for real-valued examples
- Sequences need special care (coming soon!)

Kernel Summary



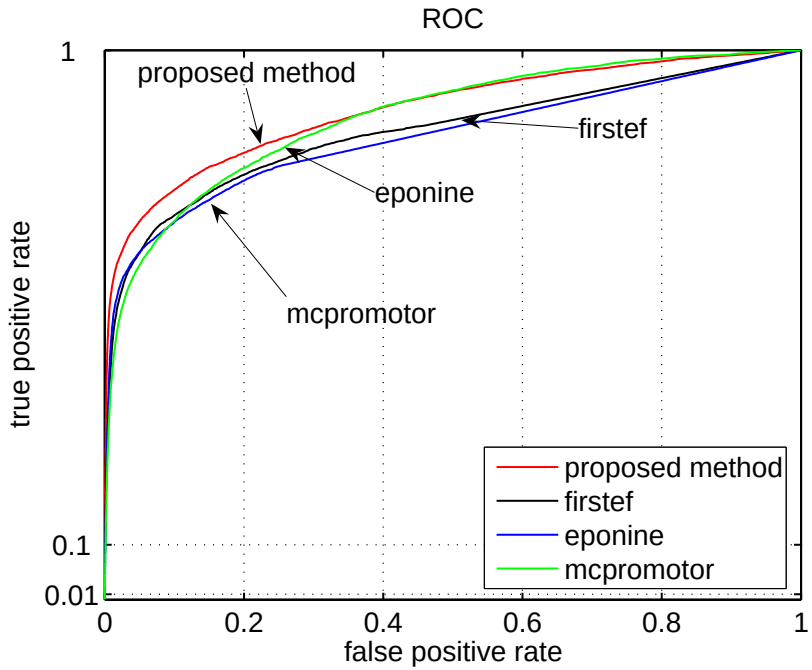
- Nonlinear separation $\Leftrightarrow$ linear separation of nonlinearly mapped examples
- Mapping Φ defines a kernel by

$$k(\mathbf{x}, \hat{\mathbf{x}}) := \langle \Phi(\mathbf{x}), \Phi(\hat{\mathbf{x}}) \rangle$$

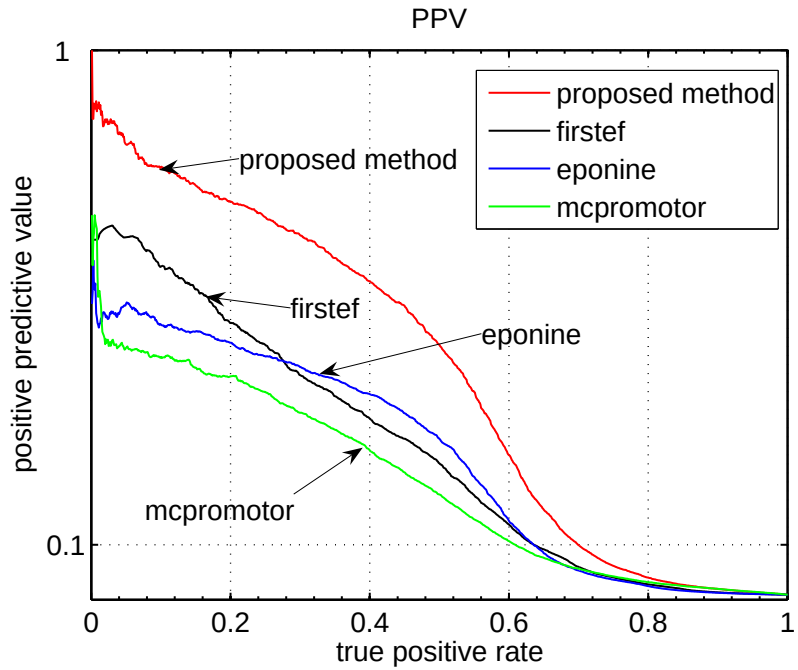
- (Mercer) Kernel defines a mapping Φ (nontrivial)
- Choice of kernel has to match the data at hand
- RBF kernel often works pretty well

Evaluation Measures for Classification I

[left] Receiver Operating Characteristic (ROC) Curve



[right] Precision Recall Curve



(Obtained by varying bias and recording TPR/FPR or PPV/TPR.)

Use bias-independent scalar evaluation measure

- Area under ROC Curve (auROC)
- Area under Precision Recall Curve (auPRC)

GC-Content-based Splice Site Recognition



Kernel	auROC
Linear	88.2%
Polynomial $d = 3$	91.4%
Polynomial $d = 7$	90.4%
Gaussian $\sigma = 100$	87.9%
Gaussian $\sigma = 1$	88.6%
Gaussian $\sigma = 0.01$	77.3%

SVM accuracy of acceptor site recognition using polynomial and Gaussian kernels with different degrees d and widths σ. Accuracy is measured using the area under the ROC curve (auROC) and is computed using five-fold cross-validation

Available SVM Packages



2-Class Classification	(35 hits on `http://mloss.org`) package names sorted by popularity
Multi-Class Classification	(7 hits on `http://mloss.org`)
Regression	(54 hits on `http://mloss.org`)

More can be found at `http://www.kernel-machines.org`.

Easy-to-use Software



Easysvm	an easy-to-use SVM toolbox based on Python and the Shogun toolbox, usable from command line or within Python
PyML	an easy-to-use Python-based SVM toolbox, usable from command line or within Python
Shogun toolbox	a powerful toolbox for large-scale data analysis, including many SVM implementations with support for Python, R, Matlab, and Octave
LibSVM	an SVM library with a graphic interface
SVM-Light	an efficient implementation of SVMs in C, usable from command line
Galaxy Web Service	a web service for using SVMs, using predefined kernels for real-valued data and string classification (based on Easysvm): `http://galaxy.fml.tuebingen.mpg.de`

The String Kernel Recipe



General idea

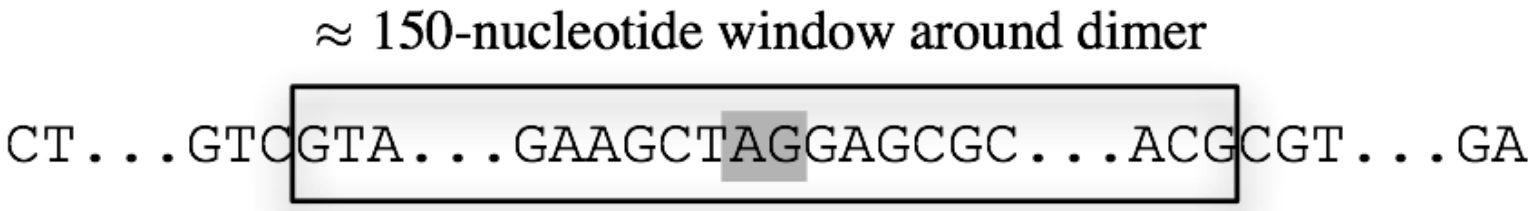
- Count substrings shared by two strings
- The greater the number of common substrings, the more two sequences are deemed similar

Variations

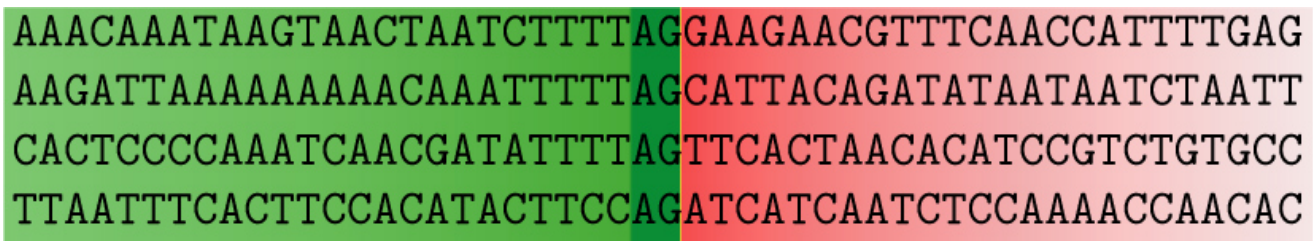
- Allow gaps or mismatches
- Include wildcards or physico-chemical properties
- Motif kernels
- Consider position in the sequence

Recognizing Genomic Signals

Discriminate true signal positions from 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



Problem categorization

(based on positional variability of motifs)

Position-Independent

→ Motifs may occur anywhere,



for instance, tissue classification using promoter region

Types of Signal Detection Problems



Problem categorization

(based on positional variability of motifs)

Position-Dependent

→ Motifs very stiff, almost always at same position,

```
AAACAAATAAGTAACTAATCTTTTAAGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAACATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAATTCACTAACACATCCGTCTGTGCC
```

for instance, splice site identification

Types of Signal Detection Problems



Problem categorization

(based on positional variability of motifs)

Mixture of Position-Dependent/-Independent

$\longrightarrow$ variable but still positional information

```
AAACAAATAAGTAACTAATCTTTTAAAGAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTCATTAAATACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTAAATTTCACTAACACATCCGTCTGTGC
```

for instance, promoter identification

Spectrum Kernel

To make use of position-independent motifs:

- **Idea:** like the bag-of-words-kernel (cf. text classification) but for biological sequences (words are now strings of length k, called k-mers)
  - Count k-mers in sequence A and sequence B.
  - Spectrum Kernel is sum of product of counts (for same k-mer)

Example $k = 3$:

x AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG

x' TACCTAATTATGAAATTAAATTTCAGTGTGCTGATGGAAACGGAGAAGTC

3-mer	AAA	AAC	...	CCA	CCC	...	TTT
# in **x**	2	4	...	1	0	...	3
# in **x**′	3	1	...	0	0	...	1

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

Weighted Degree Kernel

= Spectrum kernels for each position

To make use of position-dependent motifs:

$$k(\mathbf{x}, \mathbf{x}') = \sum_{k=1}^{d} \beta_k \sum_{l=1}^{L-k} \mathbf{I}(\mathbf{u}_{k,l}(\mathbf{x}) = \mathbf{u}_{k,l}(\mathbf{x}'))$$

- $L :=$ length of the sequence $\mathbf{x}$
- $d :=$ maximal "match length" taken into account
- $\mathbf{u}_{k,l}(\mathbf{x}) :=$ subsequence of length k at position l of sequence $\mathbf{x}$

Example degree $d = 3$:

$$k(\mathbf{x}, \mathbf{x}') = \beta_1 \cdot 21 + \beta_2 \cdot 8 + \beta_3 \cdot 4$$

[Rätsch and Sonnenburg, 2004, Sonnenburg et al., 2007a]

Weighted Degree Kernel

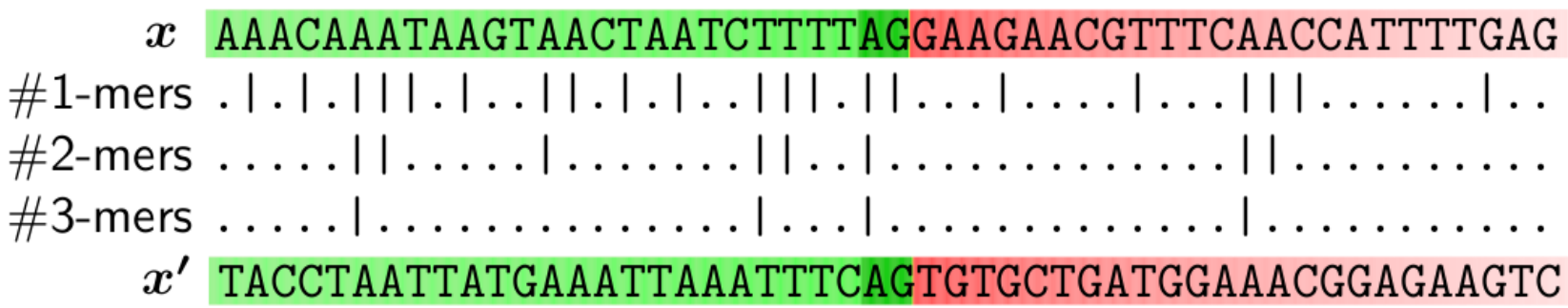
= Spectrum kernels for each position

To make use of position-dependent motifs:

$$k(\mathbf{x}, \mathbf{x}') = \sum_{k=1}^{d} \beta_k \sum_{l=1}^{L-k} \mathbf{I}(\mathbf{u}_{k,l}(\mathbf{x}) = \mathbf{u}_{k,l}(\mathbf{x}'))$$

- $L :=$ length of the sequence $\mathbf{x}$
- $d :=$ maximal "match length" taken into account
- $\mathbf{u}_{k,l}(\mathbf{x}) :=$ subsequence of length k at position l of sequence $\mathbf{x}$

Example degree $d = 3$:



$$k(\mathbf{x}, \mathbf{x}') = \beta_1 \cdot 21 + \beta_2 \cdot 8 + \beta_3 \cdot 4$$

[Rätsch and Sonnenburg, 2004, Sonnenburg et al., 2007a]

Weighted Degree Kernel

= Spectrum kernels for each position

To make use of position-dependent motifs:

$$k(\mathbf{x}, \mathbf{x}') = \sum_{k=1}^{d} \beta_k \sum_{l=1}^{L-k} \mathbf{I}(\mathbf{u}_{k,l}(\mathbf{x}) = \mathbf{u}_{k,l}(\mathbf{x}'))$$

- $L :=$ length of the sequence $\mathbf{x}$
- $d :=$ maximal "match length" taken into account
- $\mathbf{u}_{k,l}(\mathbf{x}) :=$ subsequence of length k at position l of sequence $\mathbf{x}$

Difference to Spectrum kernel:

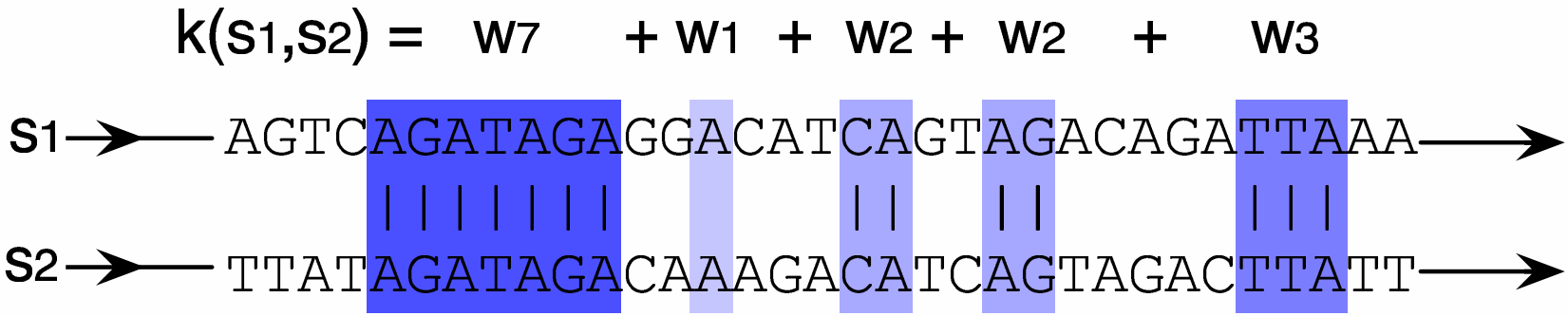
- Mixture of Spectrum kernels (up to degree d)
- Each position is considered independently

[Rätsch and Sonnenburg, 2004, Sonnenburg et al., 2007a]

Weighted Degree Kernel

- As weighting we use $\beta_k = 2\frac{d-k+1}{d(d+1)}$:
  - Longer matches are weighted less, but they imply many shorter matches
- Computational effort is $O(L \cdot d)$

Speed-up Idea: Reduce effort to $O(L)$ by finding matching "blocks" (computational effort $\mathcal{O}(L)$)



Exercise: Show that WD kernel and its "block" formulation are equivalent

Sequence-based Splice Site Recognition

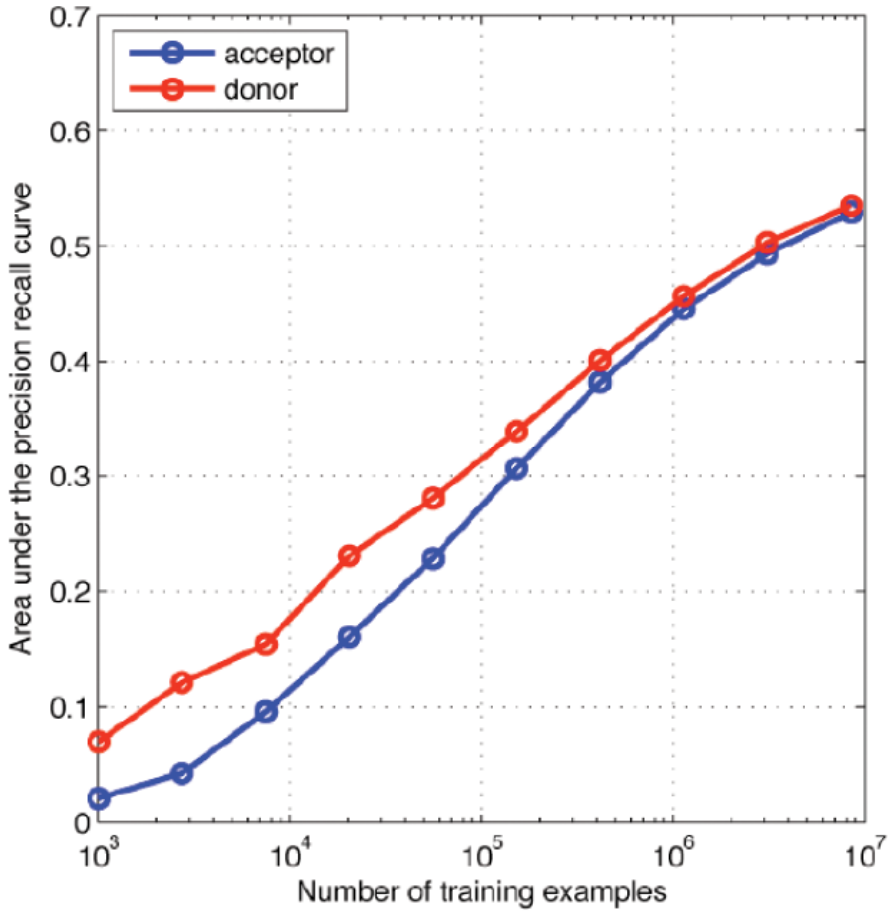


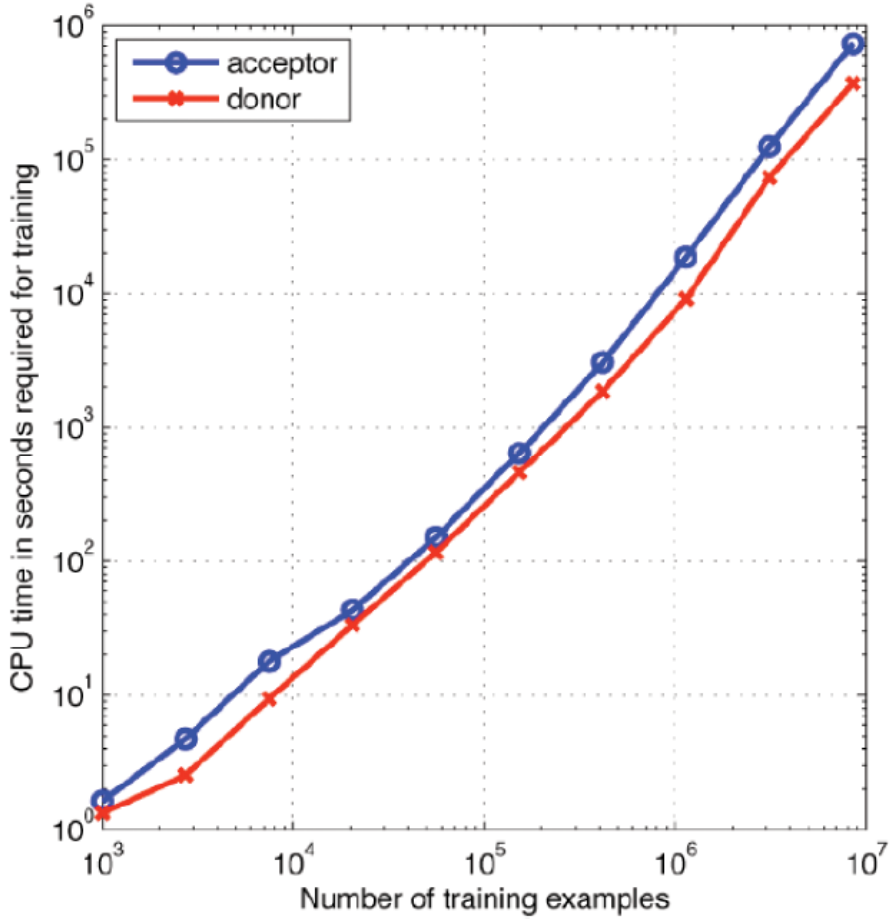
Kernel	auROC
Spectrum $\ell = 1$	94.0%
Spectrum $\ell = 3$	96.4%
Spectrum $\ell = 5$	94.5%
Mixed spectrum $\ell = 1$	94.0%
Mixed spectrum $\ell = 3$	96.9%
Mixed spectrum $\ell = 5$	97.2%
WD $\ell = 1$	98.2%
WD $\ell = 3$	98.7%
WD $\ell = 5$	98.9%

The area under the ROC curve (auROC) of SVMs with the spectrum, mixed spectrum, and weighted degree kernels for the acceptor splice site recognition task for different substring lengths ℓ.

Human Splice Sites with WD Kernel

[Sonnenburg et al., 2007b]



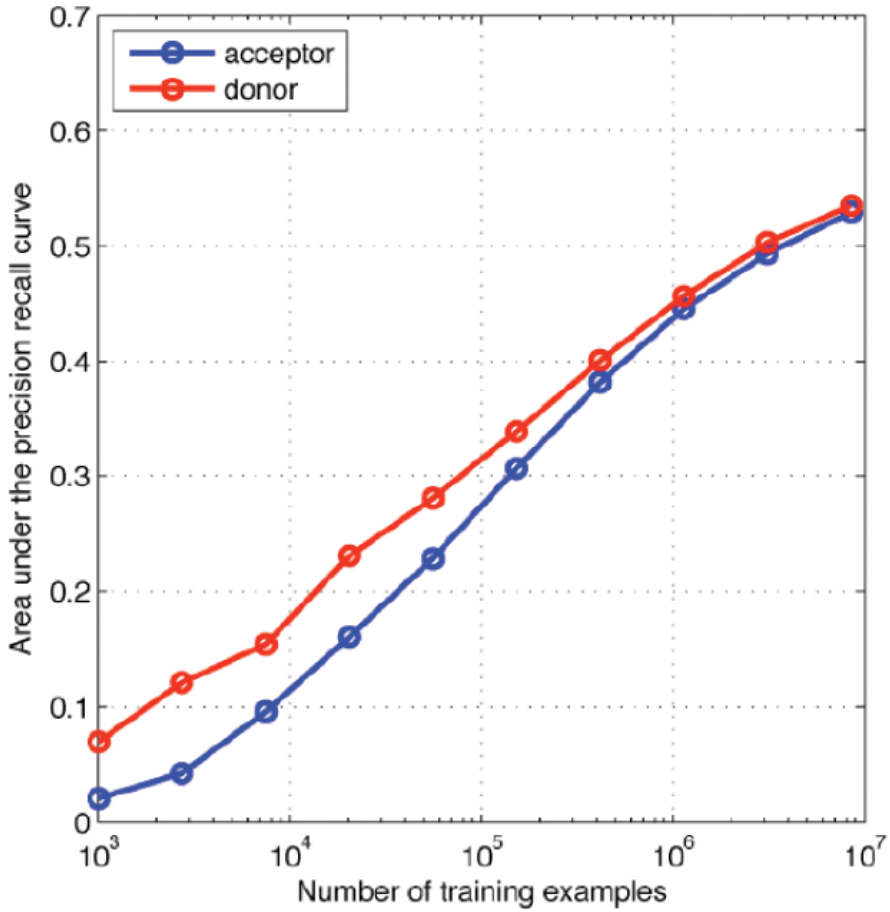


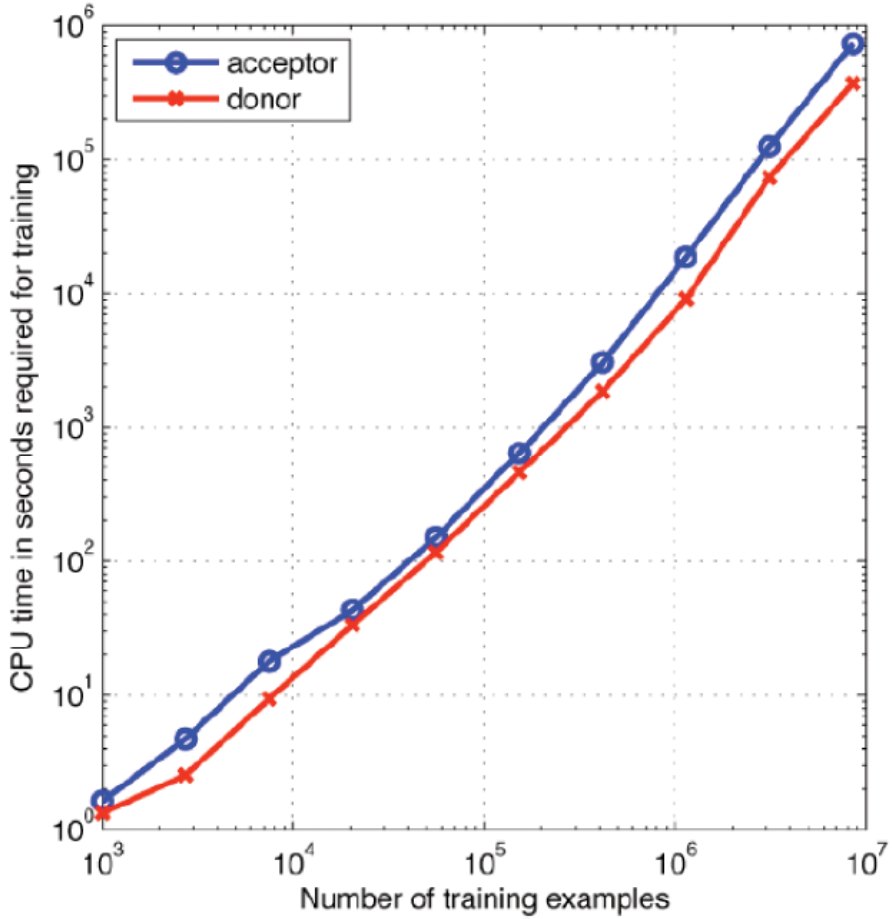
Performance (=Area under PRC curve) keeps increasing.

⇒ **Use all available data!**

Human Splice Sites with WD Kernel

[Sonnenburg et al., 2007b]





Performance (=Area under PRC curve) keeps increasing.

$\Rightarrow$ **Use all available data!**

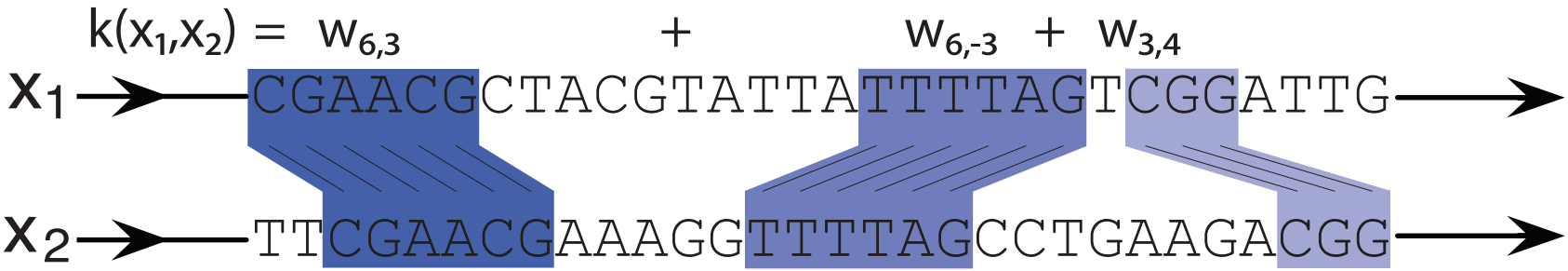
Weighted Degree Kernel with *Shifts*

To make use of partially position-dependent motifs:

- If sequence is slightly mutated (e.g. indels), WD kernel fails
- Extension: Allow some positional variance (shifts $S(l)$)

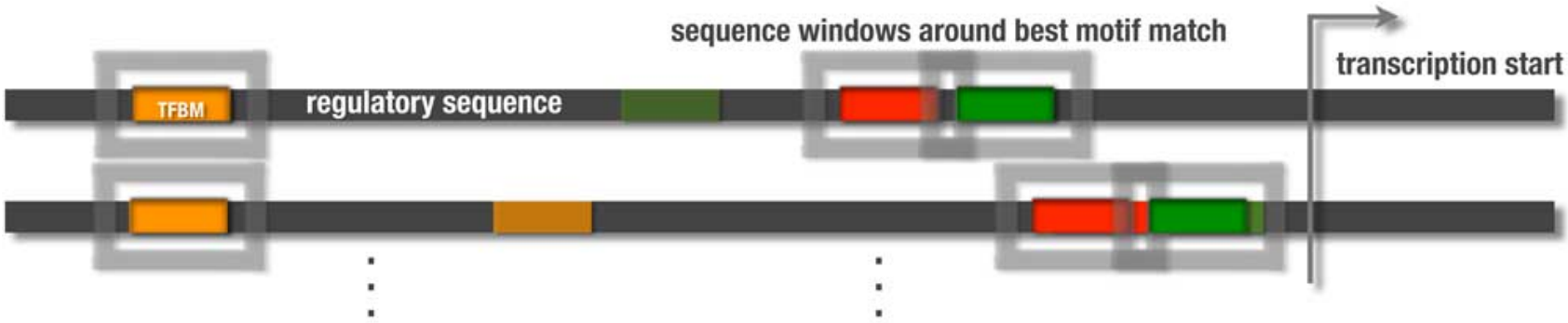
$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{K} \beta_k \sum_{l=1}^{L-k+1} \gamma_l \sum_{\substack{s=0 \\ s+l \le L}}^{S(l)} \delta_s \; \mu_{k,l,s,\mathbf{x}_i,\mathbf{x}_j},$$

$$\mu_{k,l,s,\mathbf{x}_i,\mathbf{x}_j} = \mathbf{I}(\mathbf{u}_{k,l+s}(\mathbf{x}_i) = \mathbf{u}_{k,l}(\mathbf{x}_j)) + \mathbf{I}(\mathbf{u}_{k,l}(\mathbf{x}_i) = \mathbf{u}_{k,l+s}(\mathbf{x}_j)),$$



[Rätsch et al., 2005]

Regulatory Modules Kernel [Schultheiss et al., 2008]



- Search for overrepresented motifs $m_1, \ldots, m_M$ (colored bars)
- Find best match of motif m_i in example x_j; extract windows $s_{i,j}$ at position $p_{i,j}$ around matches (boxed)
- Use a string kernel, e.g. k_{WDS}, on all extracted sequence windows, and define a combined kernel for the sequences:

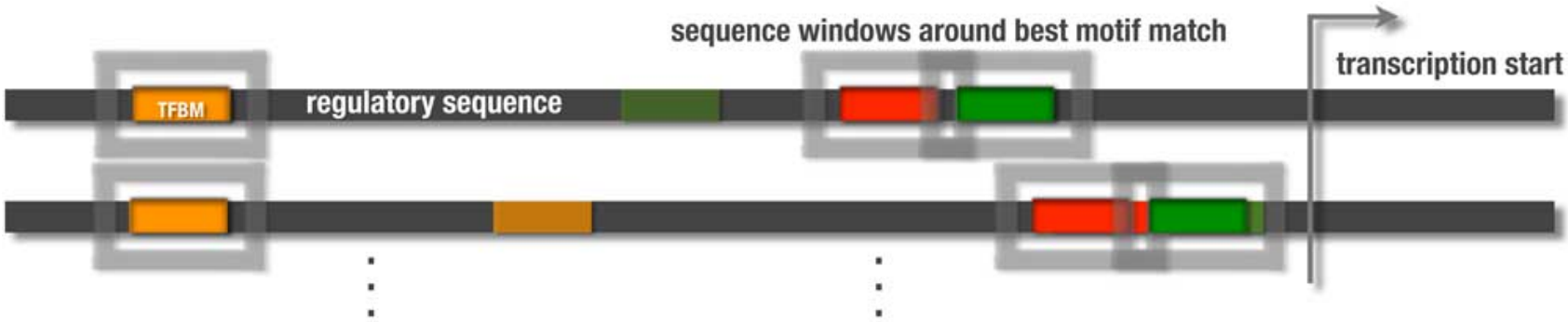
$$k_{seq}(\mathbf{x}_j, \mathbf{x}_k) = \sum_{i=1}^{M} k_{WDS}(s_{i,j}, s_{i,k})$$

- Use a second kernel k_{pos}, e.g. based on RBF kernel, on vector of pairwise distances between the motif matches:

$$\mathbf{f}_j = (p_{1,j} - p_{2,j}, p_{1,j} - p_{3,j}, \ldots, p_{M-1,j} - p_{M,j})$$

- Regulatory Modules kernel: $k_{RM}(\mathbf{x}, \mathbf{x}') := k_{seq}(\mathbf{x}, \mathbf{x}') + k_{pos}(\mathbf{x}, \mathbf{x}')$

Regulatory Modules Kernel [Schultheiss et al., 2008]



- Search for overrepresented motifs $m_1, \ldots, m_M$ (colored bars)
- Find best match of motif m_i in example x_j; extract windows $s_{i,j}$ at position $p_{i,j}$ around matches (boxed)
- Use a string kernel, e.g. k_{WDS}, on all extracted sequence windows, and define a combined kernel for the sequences:

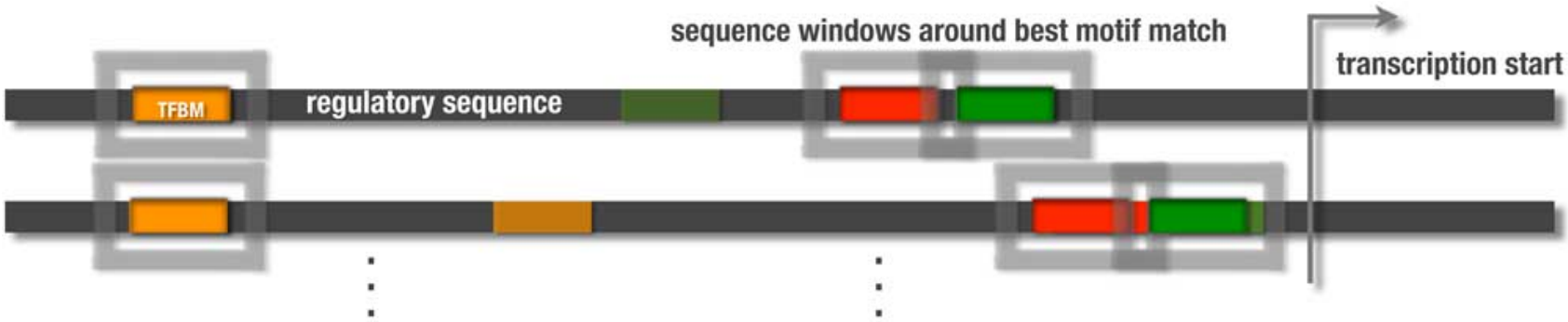
$$k_{seq}(\mathbf{x}_j, \mathbf{x}_k) = \sum_{i=1}^{M} k_{WDS}(s_{i,j}, s_{i,k})$$

- Use a second kernel k_{pos}, e.g. based on RBF kernel, on vector of pairwise distances between the motif matches:

$$\mathbf{f}_j = (p_{1,j} - p_{2,j}, p_{1,j} - p_{3,j}, \ldots, p_{M-1,j} - p_{M,j})$$

- Regulatory Modules kernel: $k_{RM}(\mathbf{x}, \mathbf{x}') := k_{seq}(\mathbf{x}, \mathbf{x}') + k_{pos}(\mathbf{x}, \mathbf{x}')$

Regulatory Modules Kernel [Schultheiss et al., 2008]



- Search for overrepresented motifs $m_1, \ldots, m_M$ (colored bars)
- Find best match of motif m_i in example x_j; extract windows $s_{i,j}$ at position $p_{i,j}$ around matches (boxed)
- Use a string kernel, e.g. k_{WDS}, on all extracted sequence windows, and define a combined kernel for the sequences:

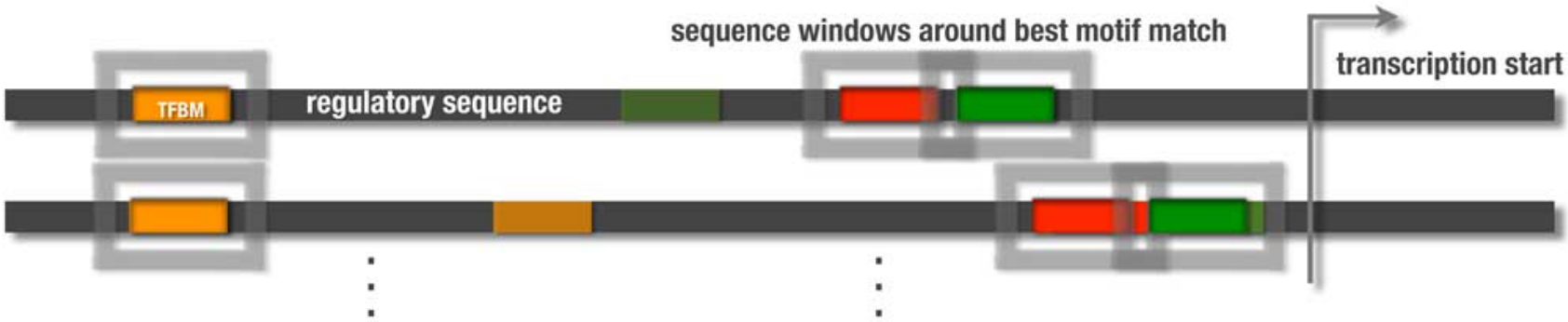
$$k_{seq}(\mathbf{x}_j, \mathbf{x}_k) = \sum_{i=1}^{M} k_{WDS}(s_{i,j}, s_{i,k})$$

- Use a second kernel k_{pos}, e.g. based on RBF kernel, on vector of pairwise distances between the motif matches:

$$\mathbf{f}_j = (p_{1,j} - p_{2,j}, p_{1,j} - p_{3,j}, \ldots, p_{M-1,j} - p_{M,j})$$

- Regulatory Modules kernel: $k_{RM}(\mathbf{x}, \mathbf{x}') := k_{seq}(\mathbf{x}, \mathbf{x}') + k_{pos}(\mathbf{x}, \mathbf{x}')$

Regulatory Modules Kernel [Schultheiss et al., 2008]



- Search for overrepresented motifs $m_1, \ldots, m_M$ (colored bars)
- Find best match of motif m_i in example x_j; extract windows $s_{i,j}$ at position $p_{i,j}$ around matches (boxed)
- Use a string kernel, e.g. k_{WDS}, on all extracted sequence windows, and define a combined kernel for the sequences:

$$k_{seq}(\mathbf{x}_j, \mathbf{x}_k) = \sum_{i=1}^{M} k_{WDS}(s_{i,j}, s_{i,k})$$

- Use a second kernel k_{pos}, e.g. based on RBF kernel, on vector of pairwise distances between the motif matches:

$$\mathbf{f}_j = (p_{1,j} - p_{2,j}, p_{1,j} - p_{3,j}, \ldots, p_{M-1,j} - p_{M,j})$$

- Regulatory Modules kernel: $k_{RM}(\mathbf{x}, \mathbf{x}') := k_{seq}(\mathbf{x}, \mathbf{x}') + k_{pos}(\mathbf{x}, \mathbf{x}')$

Summary of String Kernels



Kernel	$l_{\mathbf{x}} \neq l_{\mathbf{x}'}$	$Pr(x\|\theta)$	Positional?	Scope	Complexity
linear	no	no	yes	local	$\mathcal{O}(l_{\mathbf{x}})$
polynomial	no	no	yes	global	$\mathcal{O}(l_{\mathbf{x}})$
locality-improved	no	no	yes	local/global	$\mathcal{O}(l \cdot l_{\mathbf{x}})$
sub-sequence	yes	no	yes	global	$\mathcal{O}(n l_{\mathbf{x}} l_{\mathbf{x}'})$
n-gram/Spectrum	yes	no	no	global	$\mathcal{O}(l_{\mathbf{x}})$
WD	no	no	yes	local	$\mathcal{O}(l_{\mathbf{x}})$
WD with *shifts*	no	no	yes	local/global	$\mathcal{O}(s \cdot l_{\mathbf{x}})$
Oligo	yes	no	yes	local/global	$\mathcal{O}(l_{\mathbf{x}} l_{\mathbf{x}'})$
TOP	yes/no	yes	yes/no	local/global	depends
Fisher	yes/no	yes	yes/no	local/global	depends

Human Transcript Start Site Recognition

BIOINFORMATICS

Vol. 22 no. 14 2006, pages e472–e480
doi:10.1093/bioinformatics/btl250

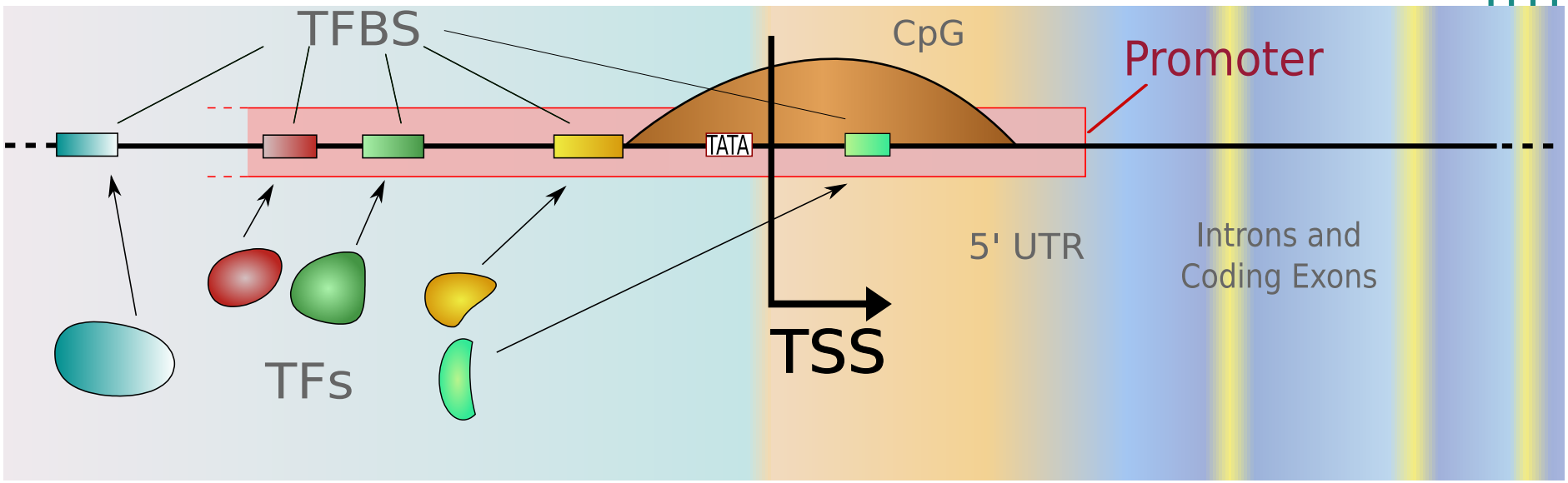
ARTS: accurate recognition of transcription starts in human

Sören Sonnenburg[1], Alexander Zien[2,3] and Gunnar Rätsch[3,*]

[1]Fraunhofer Institute FIRST, Kekuléstr. 7, Berlin, Germany, [2]Max Planck Institute for Biological Cybernetics, Spemannstr. 38, Tübingen, Germany and [3]Friedrich Miescher Laboratory, Max Planck Society, Spemannstr. 39, Tübingen, Germany

[Sonnenburg et al., 2006]

Detecting Transcription Start Sites



- POL II binds to rather unspec. 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 a 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

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

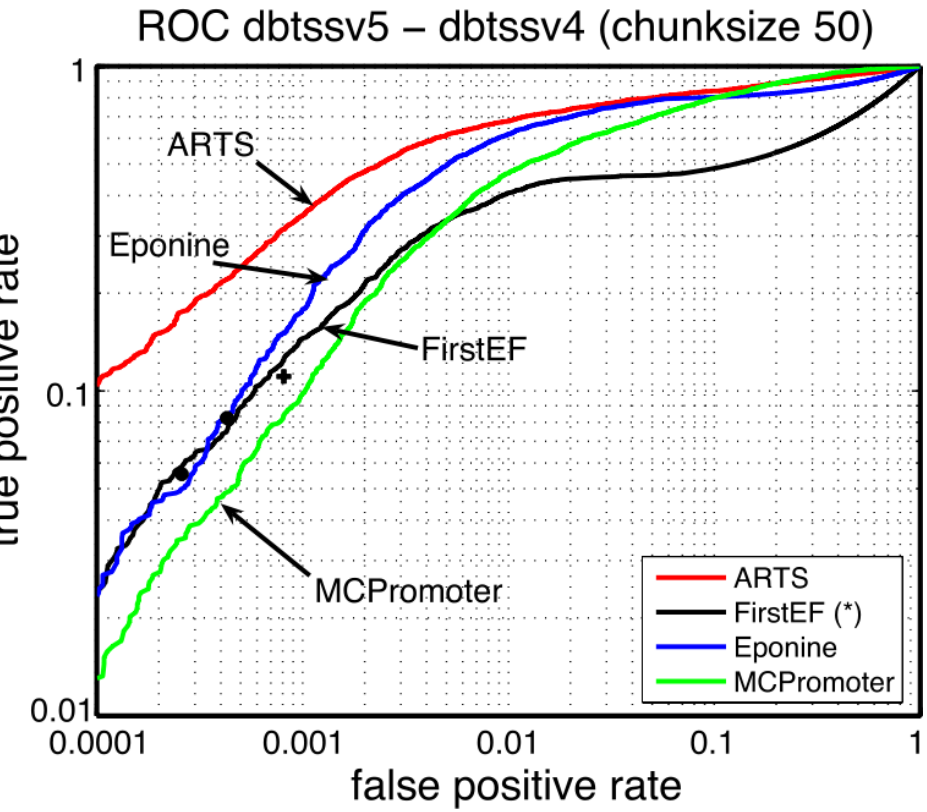
The 5 Sub-kernels

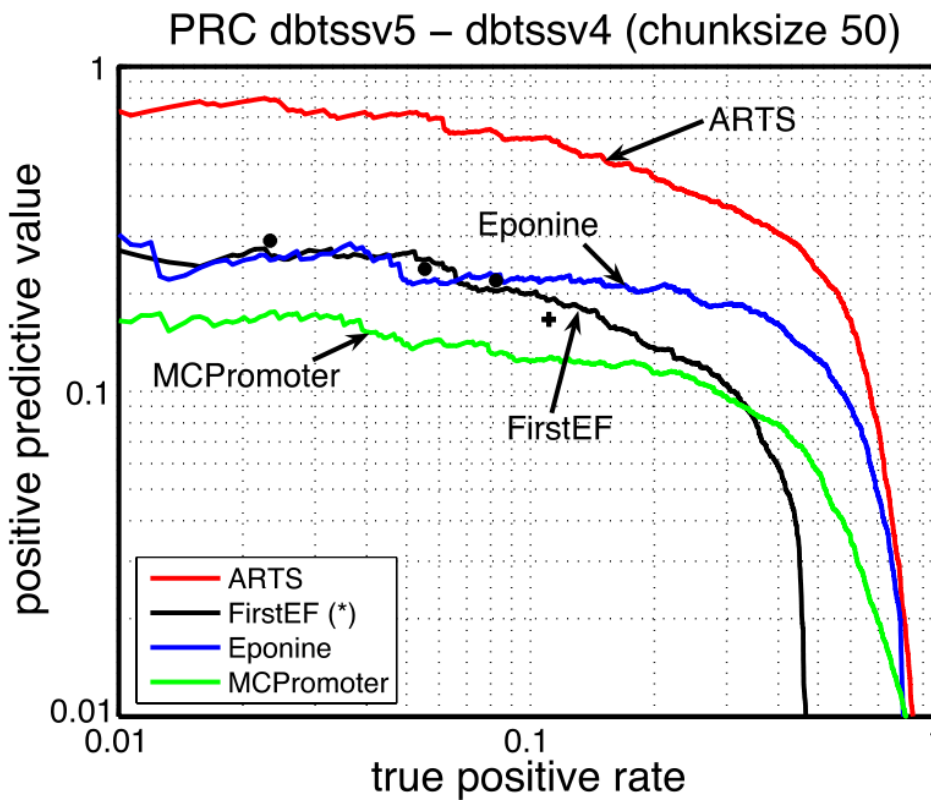


1. TSS signal (including parts of core promoter with TATA box)
   – use **Weighted Degree Shift kernel**
2. CpG Islands, distant enhancers and TFBS upstream of TSS
   – use **Spectrum kernel** (large window upstream of TSS)
3. Model coding sequence TFBS downstream of TSS
   – use another **Spectrum kernel** (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

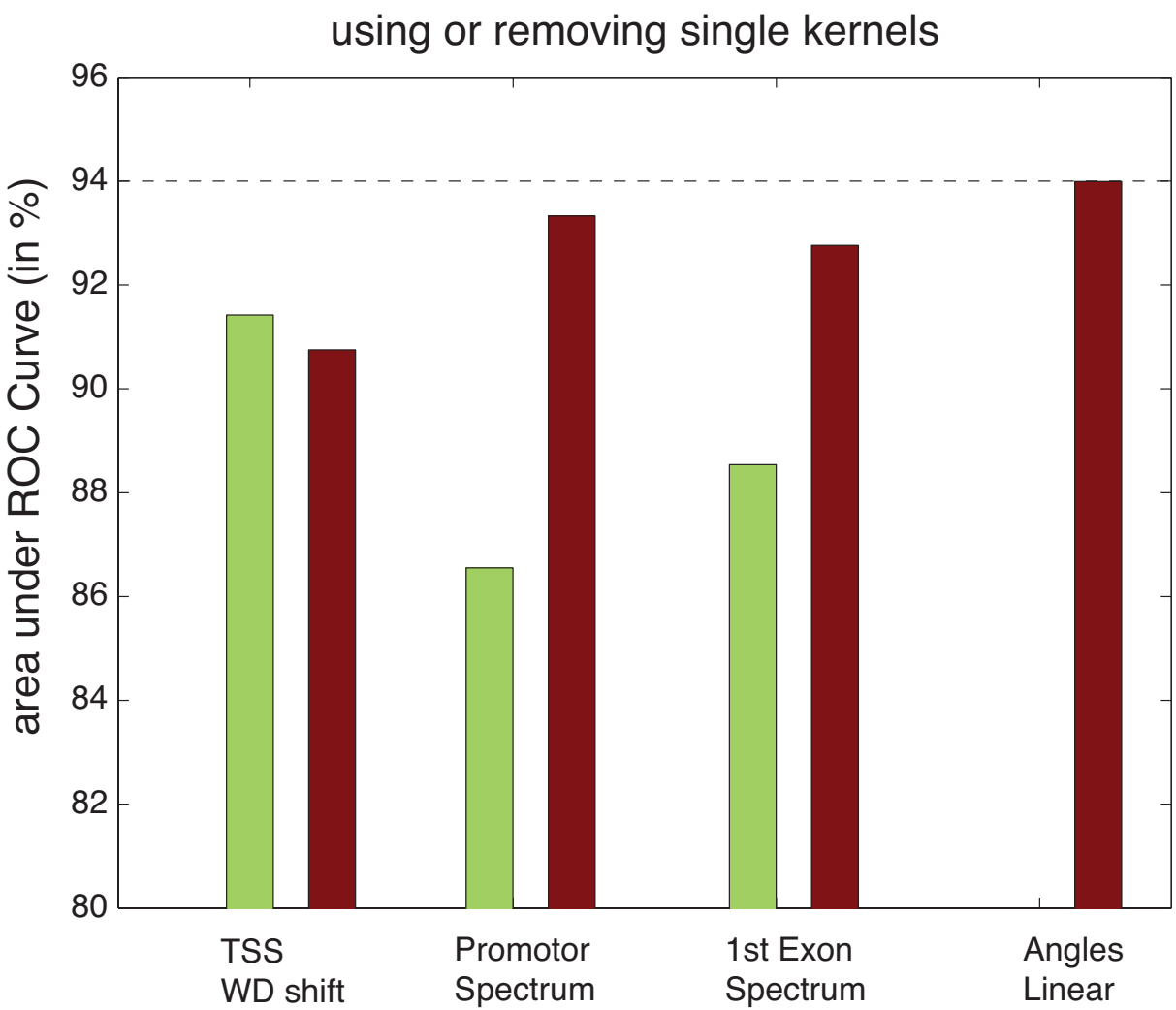
Receiver Operator Characteristic Curve and Precision Recall Curve





$\Rightarrow$ 35% **true positives at a false positive rate of** 1/1000
(best other method finds about one half (18%**))**

Contributions of the Kernels



⇒ **Most important: Weighted Degree Shift kernel modeling the TSS signal**

Hierarchical Transfer Learning

Hierarchical structure arises naturally from the Tree of Life



- Taxonomy used to define relationship between tasks
- Closer tasks benefit more from each other

Formal Problem Definition



Multitask Learning

- Consider M tasks T_i, where $i \in \{1, ..., M\}$
- We are given data $D_i = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{N_i}, y_{N_i})\}$ for each task
- We want to train M predictors $f_1, ..., f_M$, each taking into account all available information
- For that we would like to utilize a given taxonomy $\mathcal{T}$, that relates the tasks at hand

$\Rightarrow$ We need algorithms to efficiently exploit $\mathcal{T}$ for transfer learning

Formal Problem Definition



Multitask Learning

- Consider M tasks T_i, where $i \in \{1, ..., M\}$
- We are given data $D_i = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{N_i}, y_{N_i})\}$ for each task
- We want to train M predictors $f_1, ..., f_M$, each taking into account all available information
- For that we would like to utilize a given taxonomy $\mathcal{T}$, that relates the tasks at hand

$\Rightarrow$ We need algorithms to efficiently exploit $\mathcal{T}$ for transfer learning

Formal Problem Definition



Multitask Learning

- Consider M tasks T_i, where $i \in \{1, ..., M\}$
- We are given data $D_i = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_{N_i}, y_{N_i})\}$ for each task
- We want to train M predictors $f_1, ..., f_M$, each taking into account all available information
- For that we would like to utilize a given taxonomy $\mathcal{T}$, that relates the tasks at hand

$\Rightarrow$ We need algorithms to efficiently exploit $\mathcal{T}$ for transfer learning

Two ways of leveraging a given taxonomy $\mathcal{T}$



Taxonomy

Transformation

$$\Gamma = \begin{pmatrix} \gamma_{1,1} & \cdots & \gamma_{1,M} \\ \vdots & \ddots & \vdots \\ \gamma_{M,1} & \cdots & \gamma_{M,M} \end{pmatrix}$$

Task Similarity Matrix

Top-Down

Pairwise &

Multitask Kernel

Domain Adaptation by Regularization



Idea: Enforce model similarity via Regularization Term

- Regular SVM

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{(\mathbf{x},y)\in D} \ell\left(\langle \Phi(\mathbf{x}), \mathbf{w}\rangle + b, y\right)$$

- DA-SVM

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}-\mathbf{w}_{par}\|^2 + C \sum_{(\mathbf{x},y)\in D} \ell\left(\langle \Phi(\mathbf{x}), \mathbf{w}\rangle + b, y\right),$$

where ℓ is the hinge loss, $\ell(z, y) = \max\{1 - yz, 0\}$.

Domain Adaptation by Regularization



Idea: Enforce model similarity via Regularization Term

- Regular SVM

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{(\mathbf{x},y)\in D} \ell\left(\langle \Phi(\mathbf{x}), \mathbf{w}\rangle + b, y\right)$$

- DA-SVM

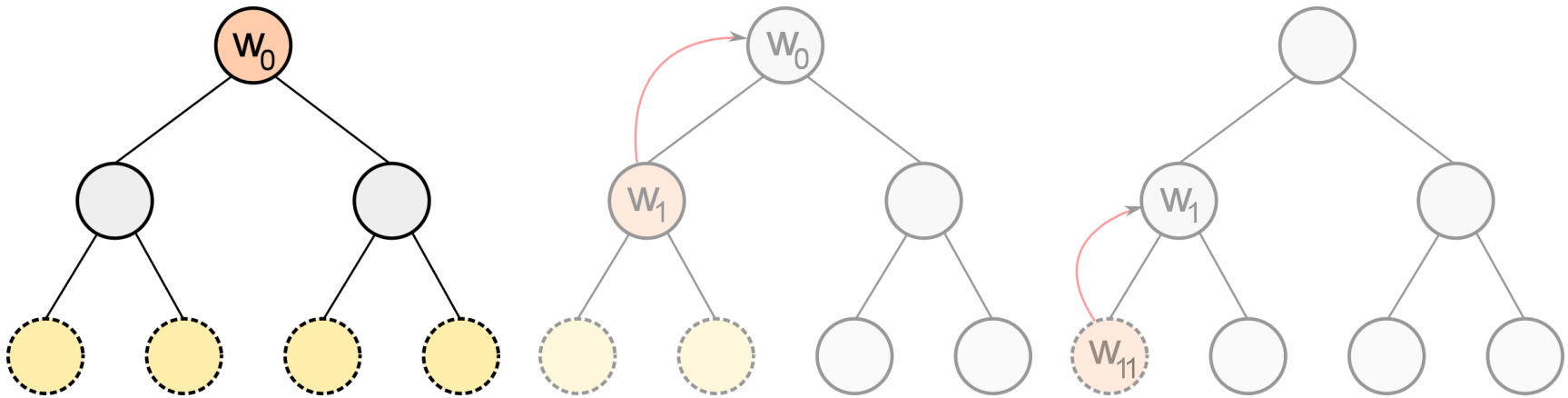
$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w} - \mathbf{w}_{par}\|^2 + C \sum_{(\mathbf{x},y)\in D} \ell\left(\langle \Phi(\mathbf{x}), \mathbf{w}\rangle + b, y\right),$$

where ℓ is the hinge loss, $\ell(z, y) = \max\{1 - yz, 0\}$.

Hierarchical Top-Down Approach

Idea: Exploit taxonomy G algorithmically

- Initialization: w_0 trained on union of task data
- Top-Down for each node t:
  - Train on $D_i = \bigcup_{j \preccurlyeq i} D_j$
  - Regularize $\mathbf{w}_i$ against parent predictor $\mathbf{w}_{par}$: $\|\mathbf{w}_i - \mathbf{w}_{par}\|^2$
- Use leaf predictors for classification



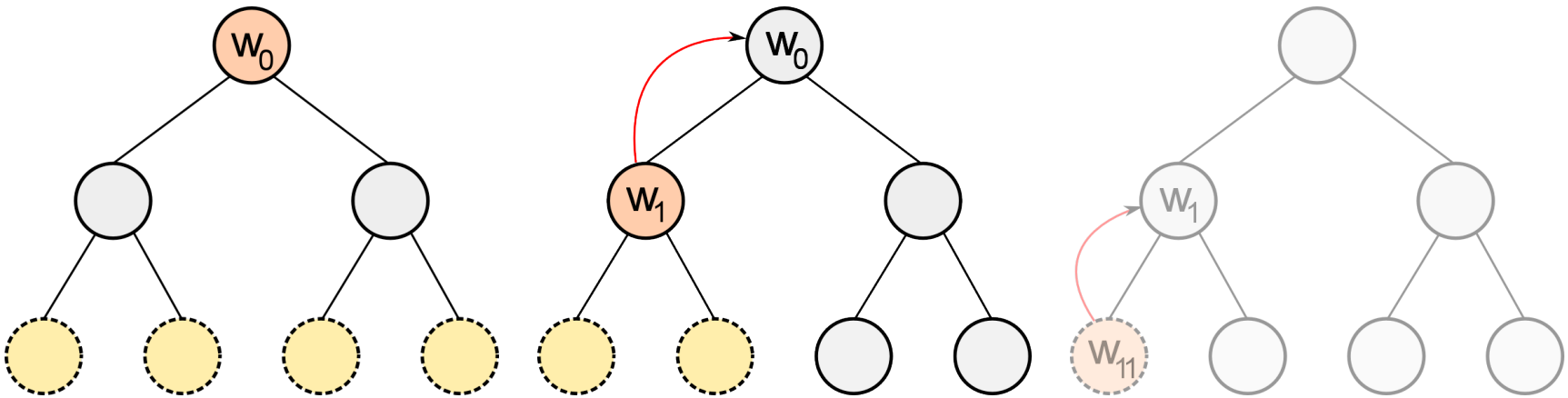
(a) Top-level training (b) Inner training (c) Taxon training

[Widmer et al., 2010]

Hierarchical Top-Down Approach

Idea: Exploit taxonomy G algorithmically

- Initialization: w_0 trained on union of task data
- Top-Down for each node t:
  - Train on $D_i = \bigcup_{j \preccurlyeq i} D_j$
  - Regularize $\mathbf{w}_i$ against parent predictor $\mathbf{w}_{par}$: $\|\mathbf{w}_i - \mathbf{w}_{par}\|^2$
- Use leaf predictors for classification



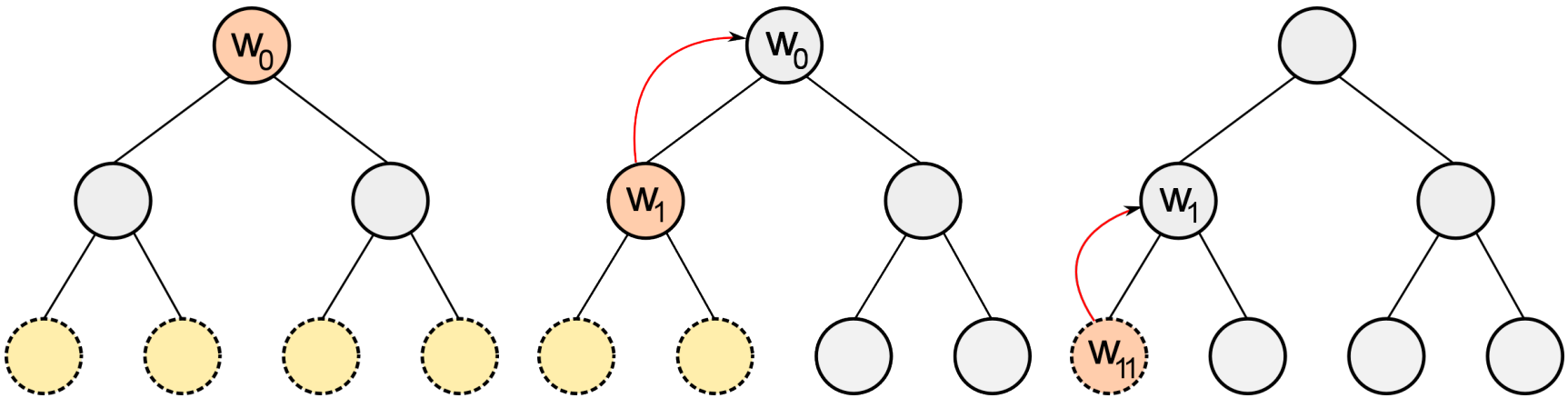
(a) Top-level training (b) Inner training (c) Taxon training

[Widmer et al., 2010]

Hierarchical Top-Down Approach

Idea: Exploit taxonomy G algorithmically

- Initialization: w_0 trained on union of task data
- Top-Down for each node t:
  - Train on $D_i = \bigcup_{j \preccurlyeq i} D_j$
  - Regularize $\mathbf{w}_i$ against parent predictor $\mathbf{w}_{par}$: $\|\mathbf{w}_i - \mathbf{w}_{par}\|^2$
- Use leaf predictors for classification



(a) Top-level training (b) Inner training (c) Taxon training

[Widmer et al., 2010]

Pairwise Approach

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_M} \quad \frac{1}{2}\sum_{t=1}^{M}\sum_{s=1}^{M}\gamma_{t,s}\|\mathbf{w}_t-\mathbf{w}_s\|^2+\sum_{t=1}^{M}C_t\sum_{(\mathbf{x},y)\in D_t}\ell\left(\langle\mathbf{x},\mathbf{w}_t\rangle,y\right).$$

where ℓ is the hinge loss, $\ell(z,y)=\max\{1-yz,0\}$.

- Train all classifiers $\mathbf{w}_i$ at the same time
- Loss is evaluated independently on datasets D_i
- Similarity is enforced via pairwise regularization term
- Task closeness controlled by $\gamma_{t,s}$

$\Rightarrow$ efficient solution possible via decomposition

[Evgeniou et al., 2005]

Multitask Kernel Approach

$$\max_{\alpha} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \hat{k}(\mathbf{x}_i,\mathbf{x}_j) + \sum_{i=1}^{n}\alpha_i$$

$$\text{s.t. } 0 \le \alpha_i \le C \;\forall i \in [1,n]$$

$$\alpha^T \mathbf{y} = 0,$$

where

$$\hat{k}((x_i,s),(x_j,t)) = \underbrace{k_{\text{task}}(s,t)}_{\gamma_{t,s}} \cdot k(x_i,x_j)$$

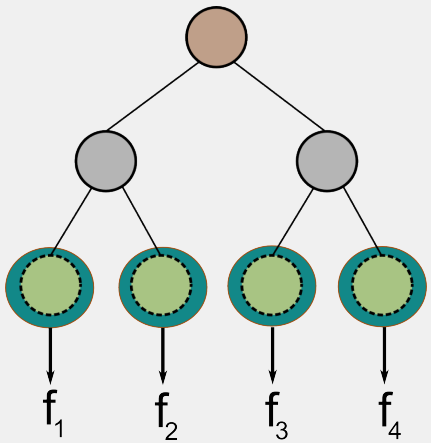
- Easily implemented by altering existing kernel functions (WDK)
- Reuse existing kernel algorithms (SVM)

[Daumé, 2007, Jacob and Vert, 2008, Widmer et al., 2010]

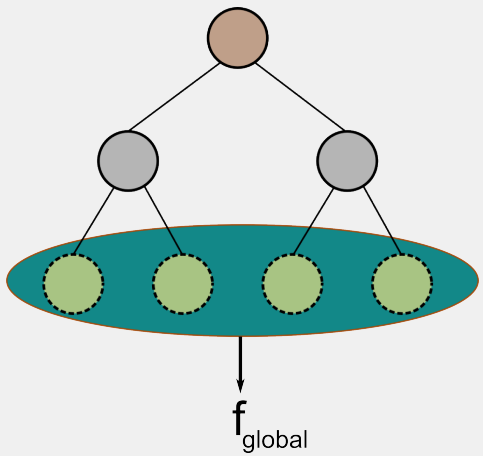
Method Summary

- Multitask Learning Methods
  - Top-Down
  - Pairwise Regularization
  - Multitask Kernel
- Additional Baselines

Plain



Union



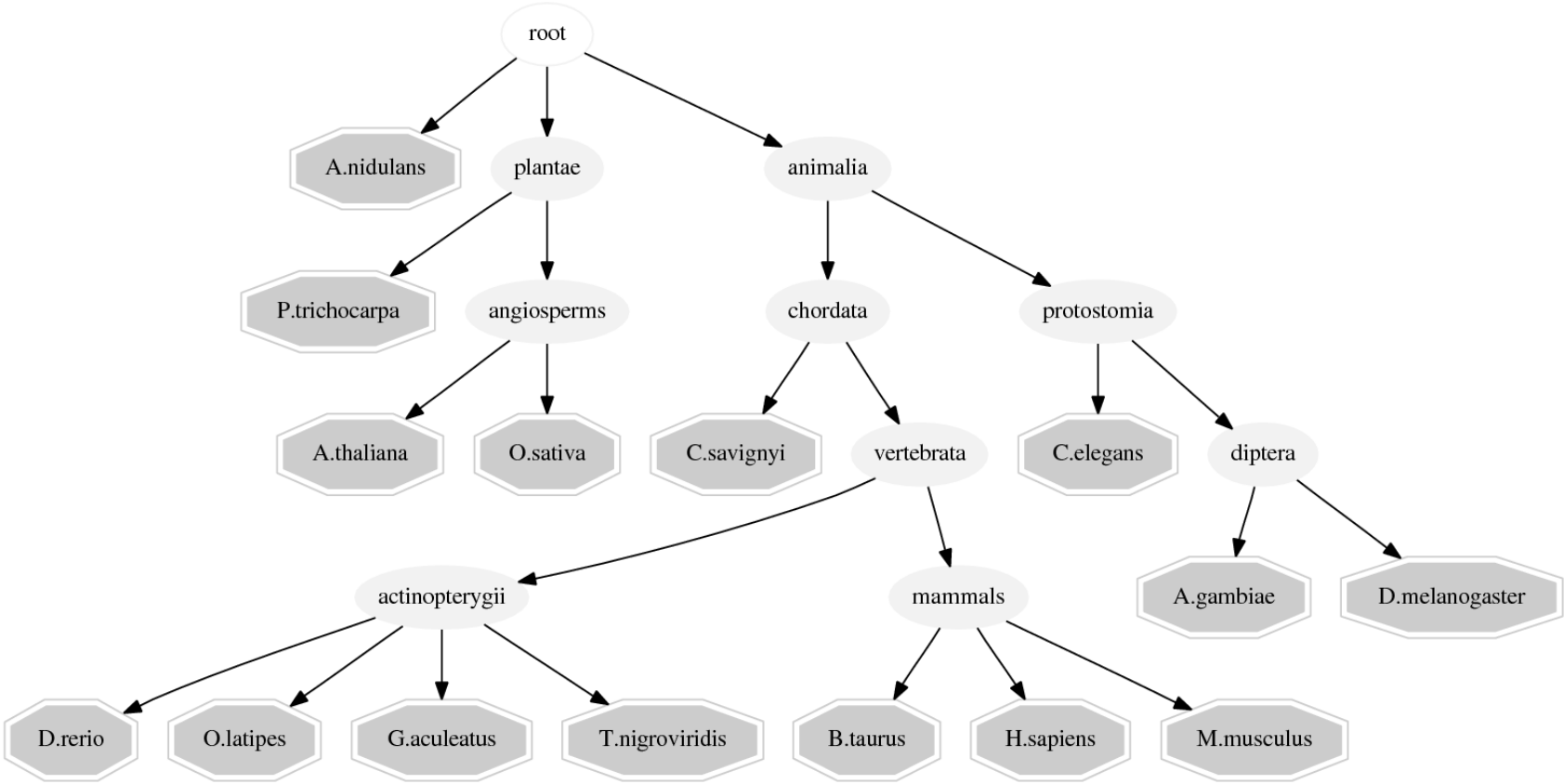
Application to splice-site recognition

- Formulate as binary classification problem

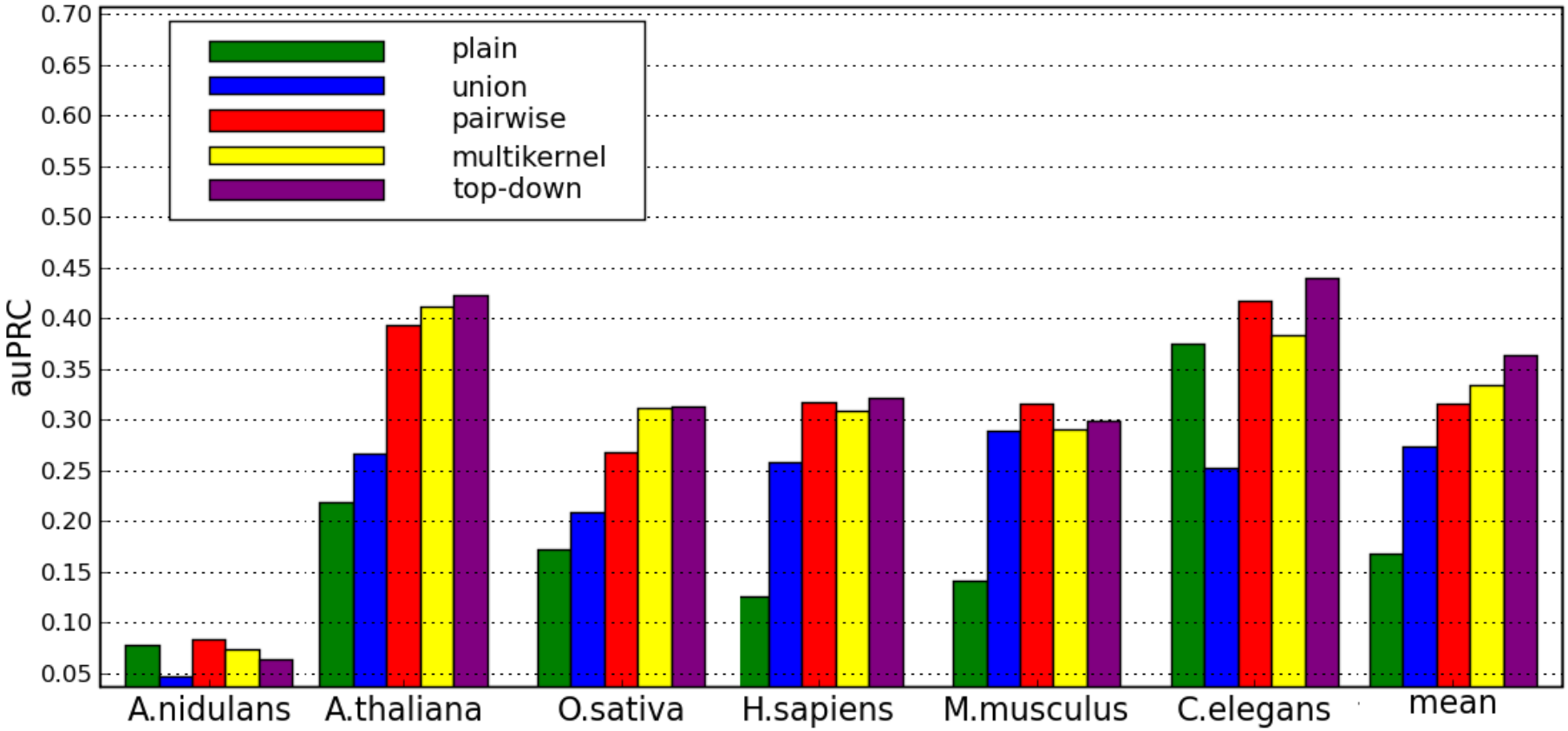
≈150 nucleotides window around dimer

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

- Utilize 15 organisms related by taxonomy

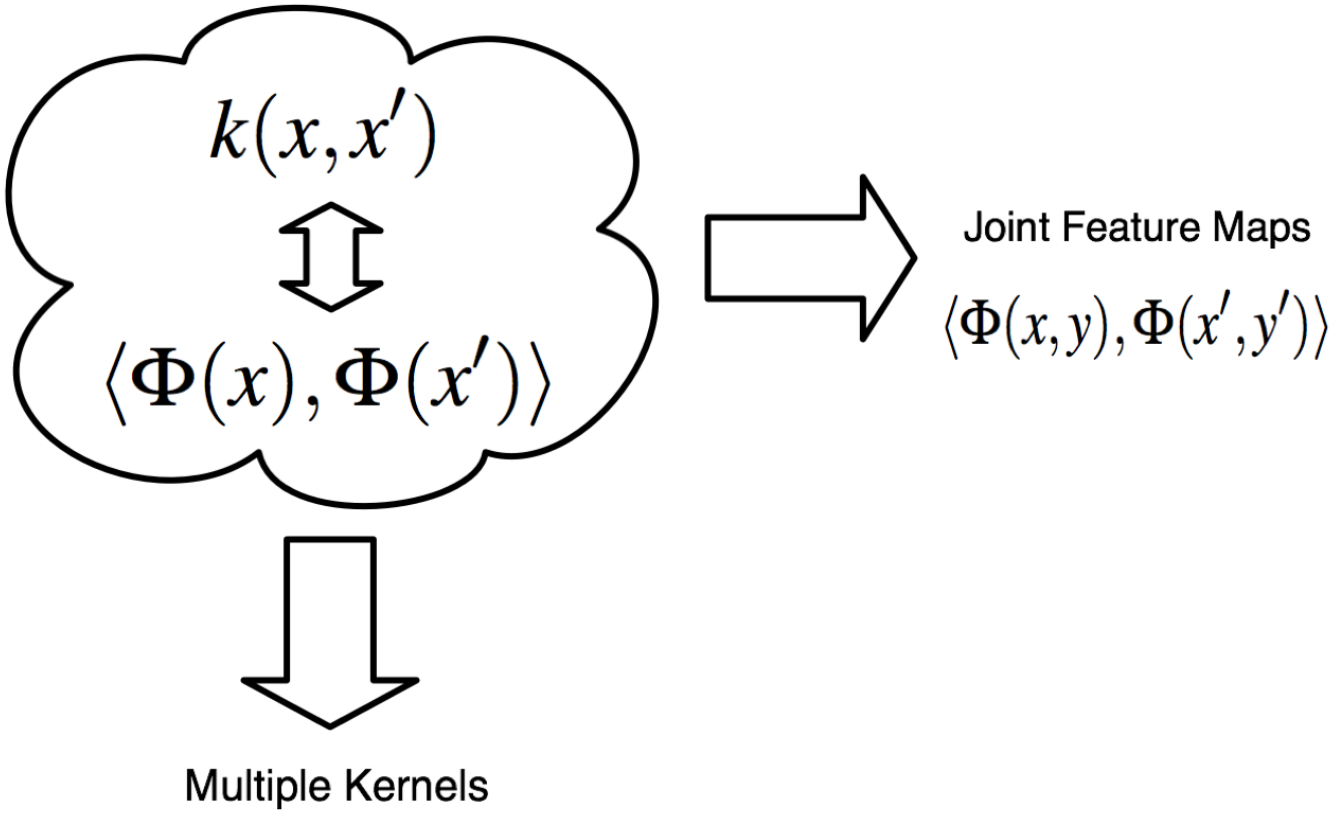


Results Splicing Data



- MTL methods outperform baselines
- Best performer is Top-Down
- Potential increase depends on the particular organism

Generalizing Kernels



- Finding the optimal combination of kernels
- **Learning structured output spaces**

Structured Output Spaces

Learning task

For a set of labeled data, we predict the label

Difference from multiclass

The set of possible labels $\mathcal{Y}$ may be very large or hierarchical

Joint kernel on $\mathcal{X}$ and $\mathcal{Y}$

We define a joint feature map on $\mathcal{X} \times \mathcal{Y}$, denoted by $\Phi(\mathbf{x}, y)$. Then the corresponding kernel function is

$$k((\mathbf{x}, y), (\mathbf{x}', y')) := \langle \Phi(\mathbf{x}, y), \Phi(\mathbf{x}', y') \rangle$$

For multiclass

For normal multiclass classification, the joint feature map decomposes and the kernels on $\mathcal{Y}$ are the identity, that is,

$$k((\mathbf{x}, y), (\mathbf{x}', y')) := [[y = y']] k(\mathbf{x}, \mathbf{x}')$$

Label Sequence Learning



- Given: observation sequence
- Problem: predict corresponding state sequence
- Often: several subsequent positions have the same state
  $\Rightarrow$ state sequence defines a "segmentation"

Label Sequence Learning

- Given: observation sequence
- Problem: predict corresponding state sequence
- Often: several subsequent positions have the same state
  $\Rightarrow$ state sequence defines a "segmentation"
- Example 1: Secondary Structure Prediction of Proteins

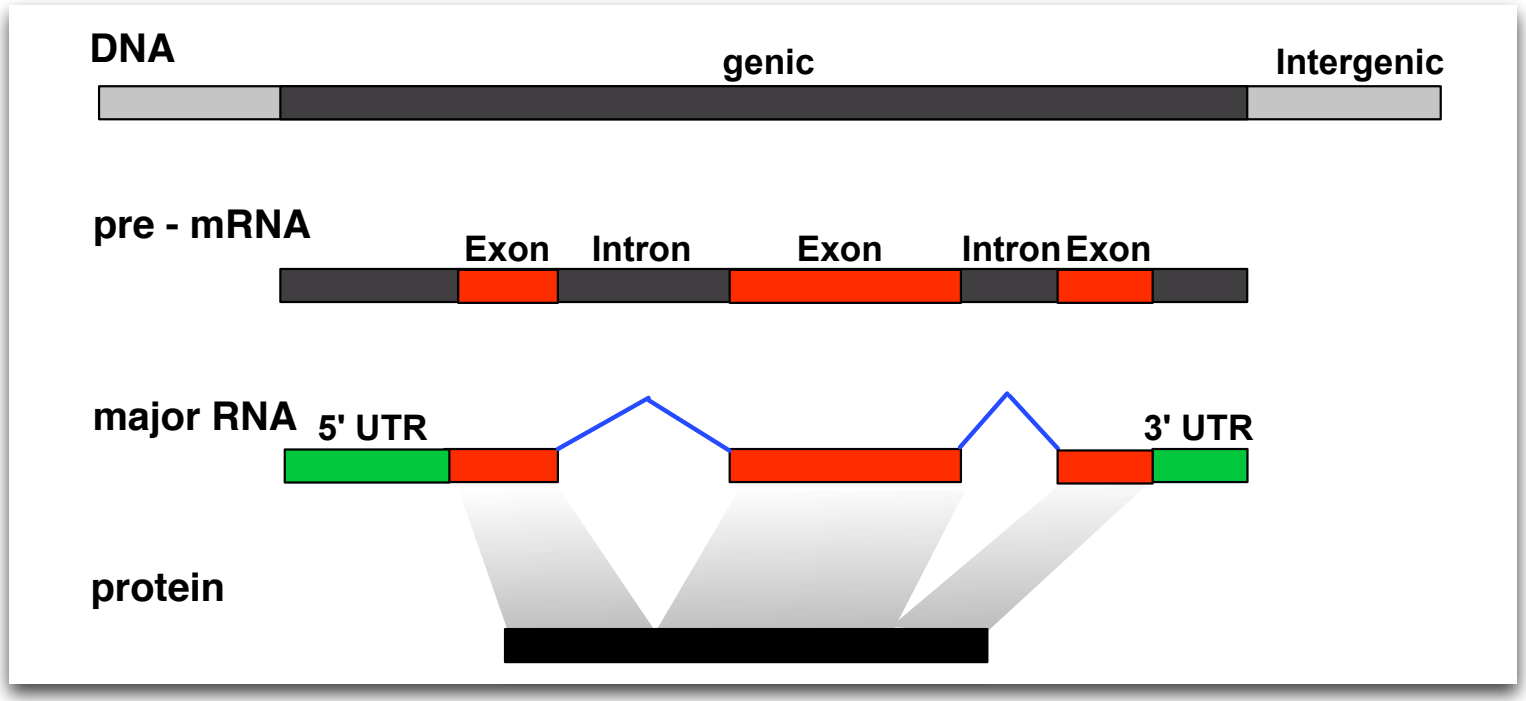


Residue Sequence: NWVLSTAADMQGVVTDGMASFL...

Secondary Structure: LLEEEELLLLHHHHHHHHHHLH...

Label Sequence Learning

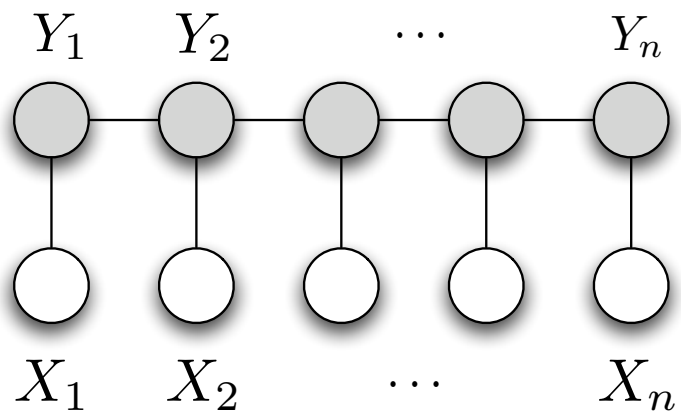
- Given: observation sequence
- Problem: predict corresponding state sequence
- Often: several subsequent positions have the same state
  $\Rightarrow$ state sequence defines a "segmentation"
- Example 2: Gene Finding



Generative Models

- Hidden Markov Models (Rabiner, 1989)
  - State sequence treated as Markov chain
  - No direct dependencies between observations
  - Example: First-order HMM (simplified)

$$p(\mathbf{x}, \mathbf{y}) = \prod_i p(x_i|y_i)p(y_i|y_{i-1})$$

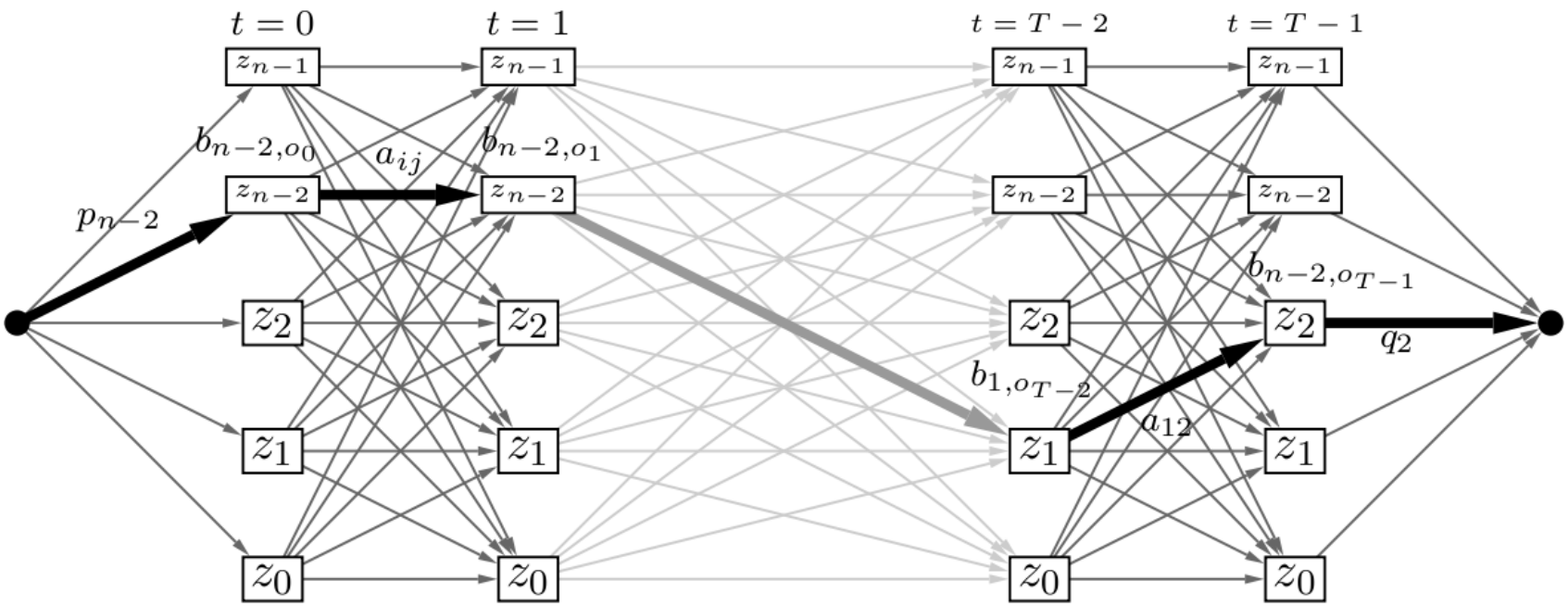


  - Efficient dynamic programming (DP) algorithms

Dynamic Programming

- Number of possible paths of length T for a (fully connected) model with n states is n^T
- Infeasible even for small T

Solution: Use dynamic programming (Viterbi decoding)

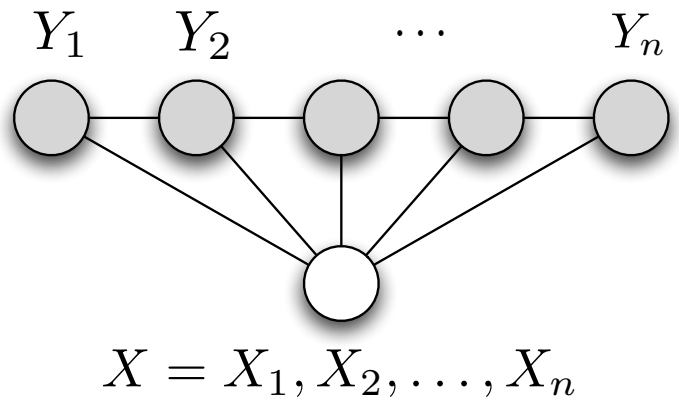


- Runtime complexity **before:** $\mathcal{O}(n^T) \Rightarrow$ **now:** $\mathcal{O}(n^2 \cdot T)$

Discriminative Models

- Conditional Random Fields [Lafferty et al., 2001]
  - Conditional probability $p(y|x)$ instead of joint probability $p(x, y)$

$$p_{\mathbf{w}}(y|x) = \frac{1}{Z(x, \mathbf{w})} \exp(f_{\mathbf{w}}(y|x))$$

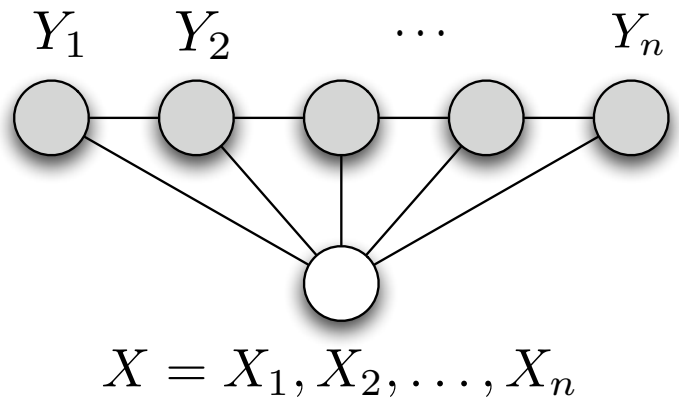


  - Can handle non-independent input features
  - Parameter estimation: $\max_{\mathbf{w}} \sum_{n=1}^{N} \log p_{\mathbf{w}}(\mathbf{y}_n|\mathbf{x}_n)$
- Decoding: Viterbi or Maximum Expected Accuracy algorithms (cf. [Gross et al., 2007])

Discriminative Models

- Conditional Random Fields [Lafferty et al., 2001]
  - Conditional probability $p(y|x)$ instead of joint probability $p(x, y)$

$$p_{\mathbf{w}}(y|x) = \frac{1}{Z(x, \mathbf{w})} \exp(f_{\mathbf{w}}(y|x))$$



  - Can handle non-independent input features
  - Parameter estimation: $\max_{\mathbf{w}} \sum_{n=1}^{N} \log p_{\mathbf{w}}(\mathbf{y}_n|\mathbf{x}_n)$
- Decoding: Viterbi or Maximum Expected Accuracy algorithms (cf. [Gross et al., 2007])

Max-Margin Structured Output Learning



- Learn function $f(\mathbf{y}|\mathbf{x})$ scoring segmentations $\mathbf{y}$ for $\mathbf{x}$
- Maximize $f(\mathbf{y}|\mathbf{x})$ w.r.t. $\mathbf{y}$ for prediction:

$$\operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}^*} f(\mathbf{y}|\mathbf{x})$$

- Idea: $f(\mathbf{y}|\mathbf{x}) \gg f(\hat{\mathbf{y}}|\mathbf{x})$ for wrong labels $\hat{\mathbf{y}} \neq \mathbf{y}$
- Approach:
  - Given: N sequence pairs $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$ for training
  - Solve:

$$\begin{aligned} \min_f \quad & C\sum_{n=1}^{N}\xi_n + \mathbf{P}[f] \\ \text{w.r.t.} \quad & f(\mathbf{y}_n|\mathbf{x}_n) - f(\mathbf{y}|\mathbf{x}_n) \geq 1-\xi_n \\ & \text{for all } \mathbf{y}_n \neq \mathbf{y} \in \mathcal{Y}^*, n = 1,\ldots,N \end{aligned}$$

  - Exponentially many constraints!

Max-Margin Structured Output Learning



- Learn function $f(\mathbf{y}|\mathbf{x})$ scoring segmentations $\mathbf{y}$ for $\mathbf{x}$
- Maximize $f(\mathbf{y}|\mathbf{x})$ w.r.t. $\mathbf{y}$ for prediction:

$$\operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}^*} f(\mathbf{y}|\mathbf{x})$$

- Idea: $f(\mathbf{y}|\mathbf{x}) \gg f(\hat{\mathbf{y}}|\mathbf{x})$ for wrong labels $\hat{\mathbf{y}} \neq \mathbf{y}$
- Approach:
  - Given: N sequence pairs $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$ for training
  - Solve:

$$\begin{aligned} \min_f \quad & C\sum_{n=1}^{N}\xi_n + \mathbf{P}[f] \\ \text{w.r.t.} \quad & f(\mathbf{y}_n|\mathbf{x}_n) - f(\mathbf{y}|\mathbf{x}_n) \geq 1-\xi_n \\ & \text{for all } \mathbf{y}_n \neq \mathbf{y} \in \mathcal{Y}^*, n = 1,\ldots,N \end{aligned}$$

  - Exponentially many constraints!

Max-Margin Structured Output Learning



- Learn function $f(\mathbf{y}|\mathbf{x})$ scoring segmentations $\mathbf{y}$ for $\mathbf{x}$
- Maximize $f(\mathbf{y}|\mathbf{x})$ w.r.t. $\mathbf{y}$ for prediction:

$$\operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}^*} f(\mathbf{y}|\mathbf{x})$$

- Idea: $f(\mathbf{y}|\mathbf{x}) \gg f(\hat{\mathbf{y}}|\mathbf{x})$ for wrong labels $\hat{\mathbf{y}} \neq \mathbf{y}$
- Approach:
  - Given: N sequence pairs $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$ for training
  - Solve:

$$\begin{aligned} \min_f \quad & C\sum_{n=1}^{N}\xi_n + \mathbf{P}[f] \\ \text{w.r.t.} \quad & f(\mathbf{y}_n|\mathbf{x}_n) - f(\mathbf{y}|\mathbf{x}_n) \geq 1-\xi_n \\ & \text{for all } \mathbf{y}_n \neq \mathbf{y} \in \mathcal{Y}^*, n = 1,\ldots,N \end{aligned}$$

  - Exponentially many constraints!

Structured Output Learning with Kernels

- Assume $f(\mathbf{y}|\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$, where $\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \in \mathcal{F}$
- Use ℓ_2 regularizer: $\mathbf{P}[f] = \|w\|^2$

$$\begin{aligned} \min_{\mathbf{w} \in \mathcal{F}, \boldsymbol{\xi} \in \mathbb{R}^N} \quad & C \sum_{n=1}^{N} \xi_n + \|w\|^2 \\ \text{w.r.t.} \quad & \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}_n) - \Phi(\mathbf{x}, \mathbf{y}) \rangle \geq 1 - \xi_n \\ & \text{for all } \mathbf{y}_n \neq \mathbf{y} \in \mathcal{Y}^*, n = 1, \ldots, N \end{aligned}$$

- Linear classifier that separates true from false labeling

Optimization

- Optimization problem too big (dual as well)

$$\begin{aligned} \min_{\mathbf{w}\in\mathcal{F},\boldsymbol{\xi}} \quad & C\sum_{n=1}^{N}\xi_n + \|w\|^2 \\ \text{w.r.t.} \quad & \langle \mathbf{w}, \Phi(\mathbf{x},\mathbf{y}_n) - \Phi(\mathbf{x},\mathbf{y})\rangle \geq 1-\xi_n \\ & \text{for all } \mathbf{y}_n \neq \mathbf{y} \in \mathcal{Y}^*, n = 1,\ldots,N \end{aligned}$$

- One constraint per example and wrong labeling
- Iterative solution
  - Begin with small set of wrong labelings
  - Solve reduced optimization problem
  - Find labelings that violate constraints
  - Add constraints, resolve
- Guaranteed Convergence

How to Find Violated Constraints?

- Constraint
$$\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}_n) - \Phi(\mathbf{x}, \mathbf{y}) \rangle \geq 1 - \xi_n$$
- Find labeling $\mathbf{y}$ that maximizes
$$\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$$
- Use dynamic programming decoding
$$\mathbf{y} = \underset{\mathbf{y} \in \mathcal{Y}^*}{\operatorname{argmax}} \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$$
(DP only works if Φ has a certain decomposition structure)
- If $\mathbf{y} = \mathbf{y}_n$, then compute second best labeling as well
- If constraint is violated, then add to optimization problem

A Structured Output Algorithm

1. $\mathcal{Y}_n^1 = \emptyset$, for $n = 1, \ldots, N$
2. Solve

$$(\mathbf{w}^t, \boldsymbol{\xi}^t) = \underset{\mathbf{w}\in\mathcal{F},\boldsymbol{\xi}}{\operatorname{argmin}} \quad C\sum_{n=1}^{N}\xi_n + \|w\|^2$$

$$\text{w.r.t.} \quad \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}_n) - \Phi(\mathbf{x}, \mathbf{y})\rangle \geq 1 - \xi_n$$

$$\text{for all } \mathbf{y}_n \neq \mathbf{y} \in \mathcal{Y}_n^t, n = 1, \ldots, N$$

3. Find violated constraints ($n = 1, \ldots, N$)

$$\mathbf{y}_n^t = \underset{\mathbf{y}_n\neq\mathbf{y}\in\mathcal{Y}^*}{\operatorname{argmax}} \langle \mathbf{w}^t, \Phi(\mathbf{x}, \mathbf{y})\rangle$$

   If $\langle \mathbf{w}^t, \Phi(\mathbf{x}, \mathbf{y}_n) - \Phi(\mathbf{x}, \mathbf{y}_n^t)\rangle < 1 - \xi_n^t$, set $\mathcal{Y}_n^{t+1} = \mathcal{Y}_n^t \cup \{\mathbf{y}_n^t\}$
4. If violated constraint exists then go to 2
5. Otherwise terminate $\Rightarrow$ optimal solution

Loss Functions

- So far, 0/1-loss with slacks: If $\mathbf{y} \neq \mathbf{y}$, then prediction is wrong, but it does not matter how wrong
- Introduce loss function on labelings $\ell(\mathbf{y}, \mathbf{y}')$, e.g.
  - How many segments are wrong or missing
  - How different are the segments, etc.
- Extend optimization problem (Margin rescaling):

$$\min_{w \in \mathcal{F}, \xi} \quad C \sum_{n=1}^{N} \xi_n + \|w\|^2$$

$$\text{w.r.t.} \quad \langle \mathbf{w}, \Phi(\mathbf{x}_n, \mathbf{y}_n) - \Phi(\mathbf{x}_n, \mathbf{y}) \rangle \geq \ell(\mathbf{y}_n, \mathbf{y}) - \xi_n$$

$$\text{for all } \mathbf{y}_n \neq \mathbf{y} \in \mathcal{Y}^*, n = 1, \ldots, N$$

- Find violated constraints ($n = 1, \ldots, N$)

$$\mathbf{y}_n^t = \operatorname*{argmax}_{\mathbf{y}_n \neq \mathbf{y} \in \mathcal{Y}^*} \left(\langle \mathbf{w}^t, \Phi(\mathbf{x}_n, \mathbf{y}) \rangle + \ell(\mathbf{y}, \mathbf{y}_n) \right)$$

Loss Functions

- So far, 0/1-loss with slacks: If $\mathbf{y} \neq \mathbf{y}$, then prediction is wrong, but it does not matter how wrong
- Introduce loss function on labelings $\ell(\mathbf{y}, \mathbf{y}')$, e.g.
  - How many segments are wrong or missing
  - How different are the segments, etc.
- Extend optimization problem (Margin rescaling):

$$\begin{aligned} \min_{\mathbf{w}\in\mathcal{F},\boldsymbol{\xi}} \quad & C\sum_{n=1}^{N}\xi_n + \|w\|^2 \\ \text{w.r.t.} \quad & \langle \mathbf{w}, \Phi(\mathbf{x}_n, \mathbf{y}_n) - \Phi(\mathbf{x}_n, \mathbf{y})\rangle \geq \ell(\mathbf{y}_n, \mathbf{y}) - \xi_n \\ & \text{for all } \mathbf{y}_n \neq \mathbf{y} \in \mathcal{Y}^*, n = 1, \ldots, N \end{aligned}$$

- Find violated constraints ($n = 1, \ldots, N$)

$$\mathbf{y}_n^t = \operatorname*{argmax}_{\mathbf{y}_n\neq\mathbf{y}\in\mathcal{Y}^*} \left(\langle \mathbf{w}^t, \Phi(\mathbf{x}_n, \mathbf{y})\rangle + \ell(\mathbf{y}, \mathbf{y}_n)\right)$$

Loss Functions

- So far, 0/1-loss with slacks: If $\mathbf{y} \neq \mathbf{y}$, then prediction is wrong, but it does not matter how wrong
- Introduce loss function on labelings $\ell(\mathbf{y}, \mathbf{y}')$, e.g.
  - How many segments are wrong or missing
  - How different are the segments, etc.
- Extend optimization problem (Margin rescaling):

$$\begin{aligned} \min_{\mathbf{w}\in\mathcal{F},\boldsymbol{\xi}} \quad & C\sum_{n=1}^{N}\xi_n + \|w\|^2 \\ \text{w.r.t.} \quad & \langle \mathbf{w}, \Phi(\mathbf{x}_n, \mathbf{y}_n) - \Phi(\mathbf{x}_n, \mathbf{y})\rangle \geq \ell(\mathbf{y}_n, \mathbf{y}) - \xi_n \\ & \text{for all } \mathbf{y}_n \neq \mathbf{y} \in \mathcal{Y}^*, n = 1, \ldots, N \end{aligned}$$

- Find violated constraints ($n = 1, \ldots, N$)

$$\mathbf{y}_n^t = \operatorname*{argmax}_{\mathbf{y}_n\neq\mathbf{y}\in\mathcal{Y}^*} \left(\langle \mathbf{w}^t, \Phi(\mathbf{x}_n, \mathbf{y})\rangle + \ell(\mathbf{y}, \mathbf{y}_n)\right)$$

Problems

- Like the kernel, the loss-function is problem specific (& crucial)
- Optimization may require many iterations
- Number of variables increases linearly
- When using kernels, solving optimization problems can become infeasible
- Evaluation of $\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$ in dynamic programming can be very expensive
  - Optimization and decoding become too expensive
- Approximation algorithms useful
- Decompose problem
  - First part uses kernels, can be pre-computed
  - Second part without kernels and only combines ingredients

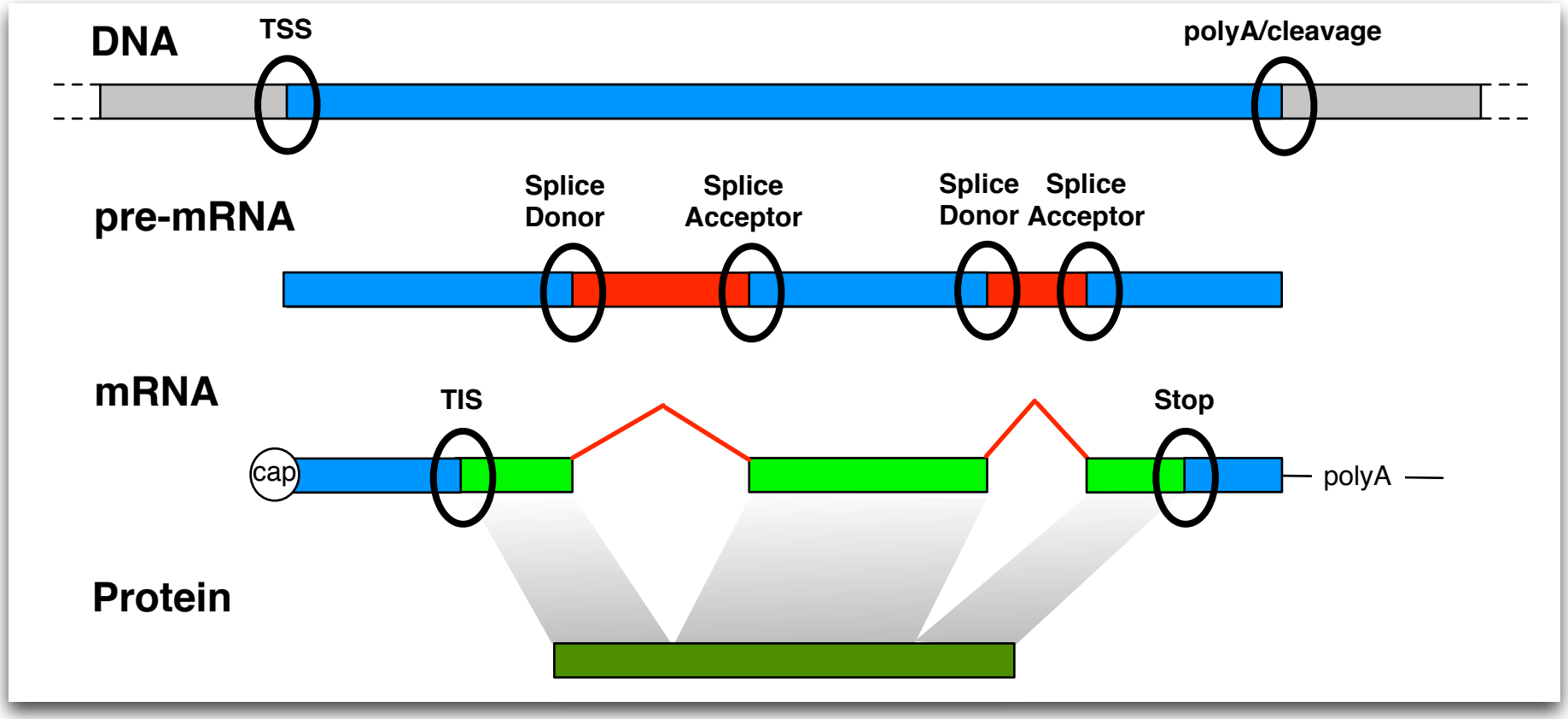
Problems

- Like the kernel, the loss-function is problem specific (& crucial)
- Optimization may require many iterations
- Number of variables increases linearly
- When using kernels, solving optimization problems can become infeasible
- Evaluation of $\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$ in dynamic programming can be very expensive
  - Optimization and decoding become too expensive
- Approximation algorithms useful
- Decompose problem
  - First part uses kernels, can be pre-computed
  - Second part without kernels and only combines ingredients

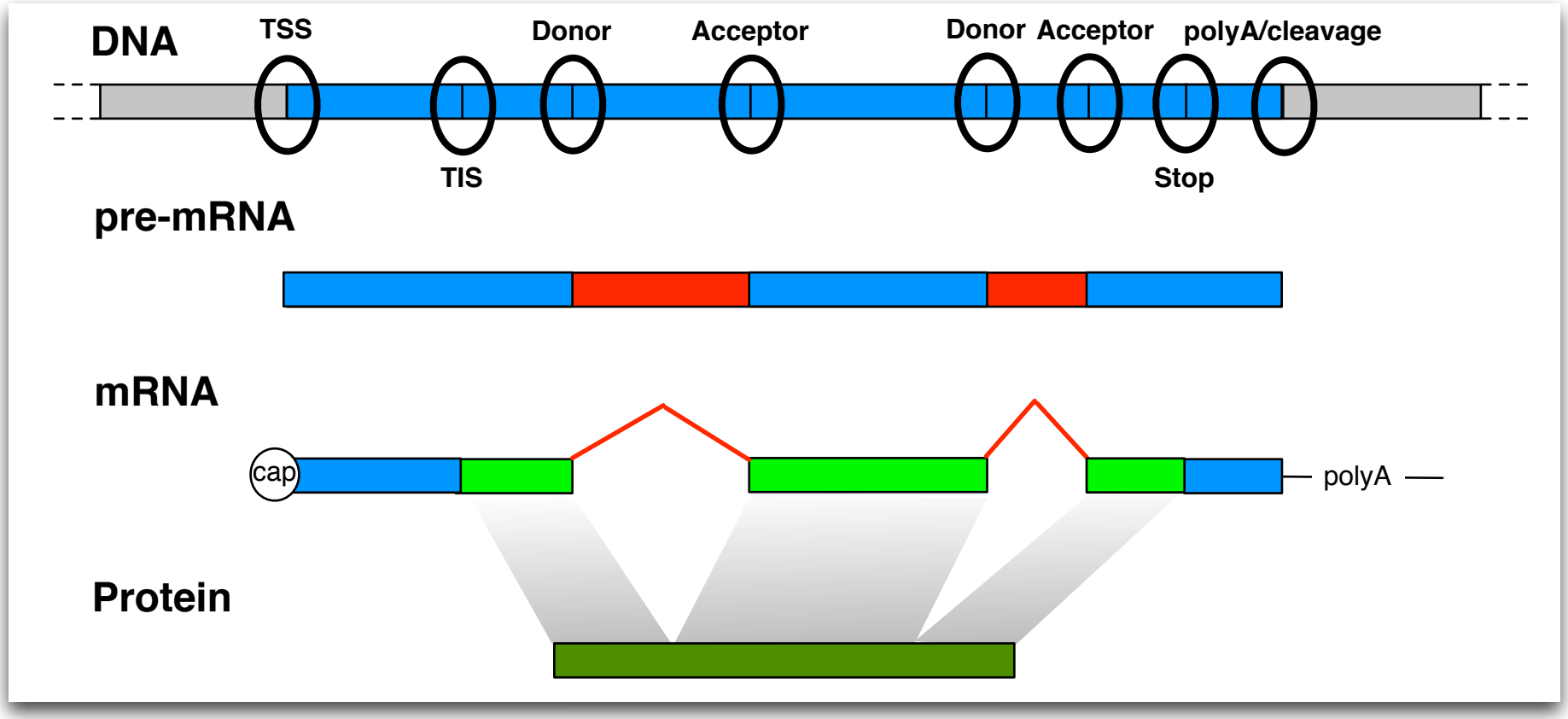
Problems

- Like the kernel, the loss-function is problem specific (& crucial)
- Optimization may require many iterations
- Number of variables increases linearly
- When using kernels, solving optimization problems can become infeasible
- Evaluation of $\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$ in dynamic programming can be very expensive
  - Optimization and decoding become too expensive
- Approximation algorithms useful
- Decompose problem
  - First part uses kernels, can be pre-computed
  - Second part without kernels and only combines ingredients

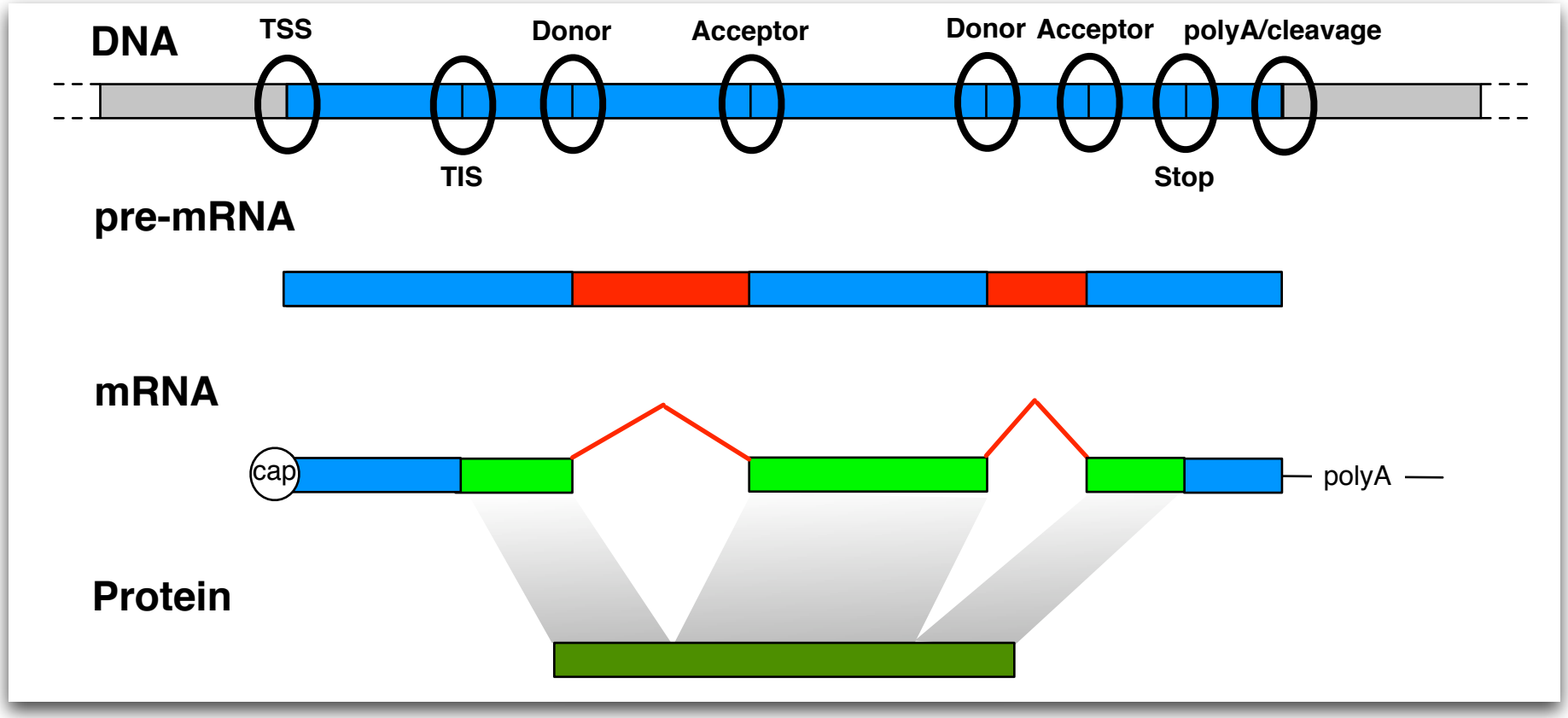
Back to Gene Finding

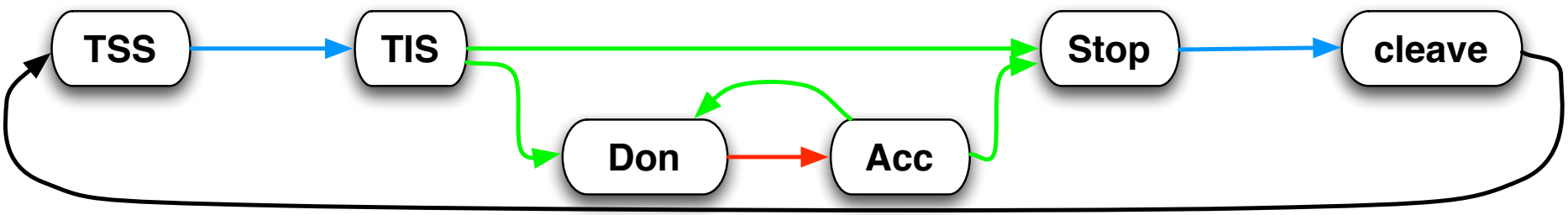


Back to Gene Finding

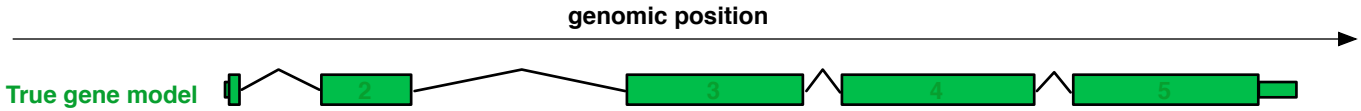


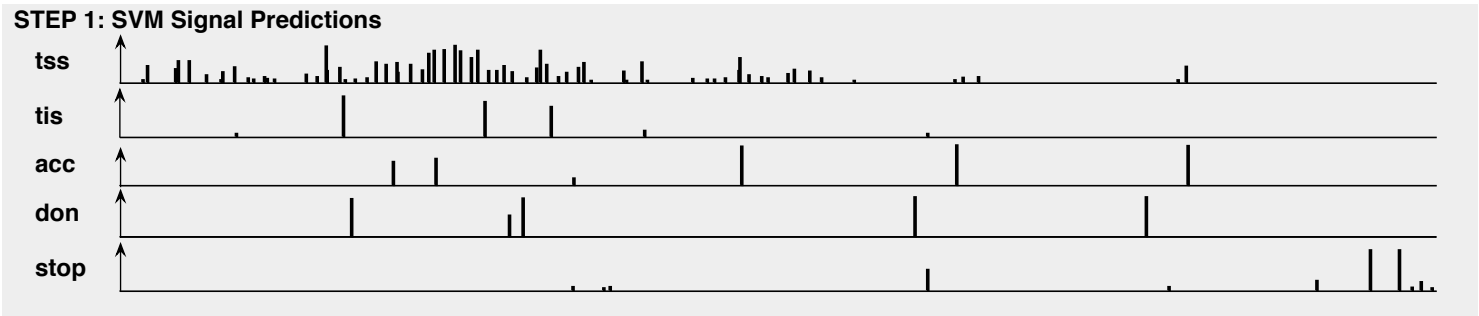
Back to Gene Finding

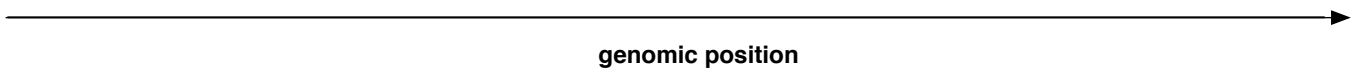




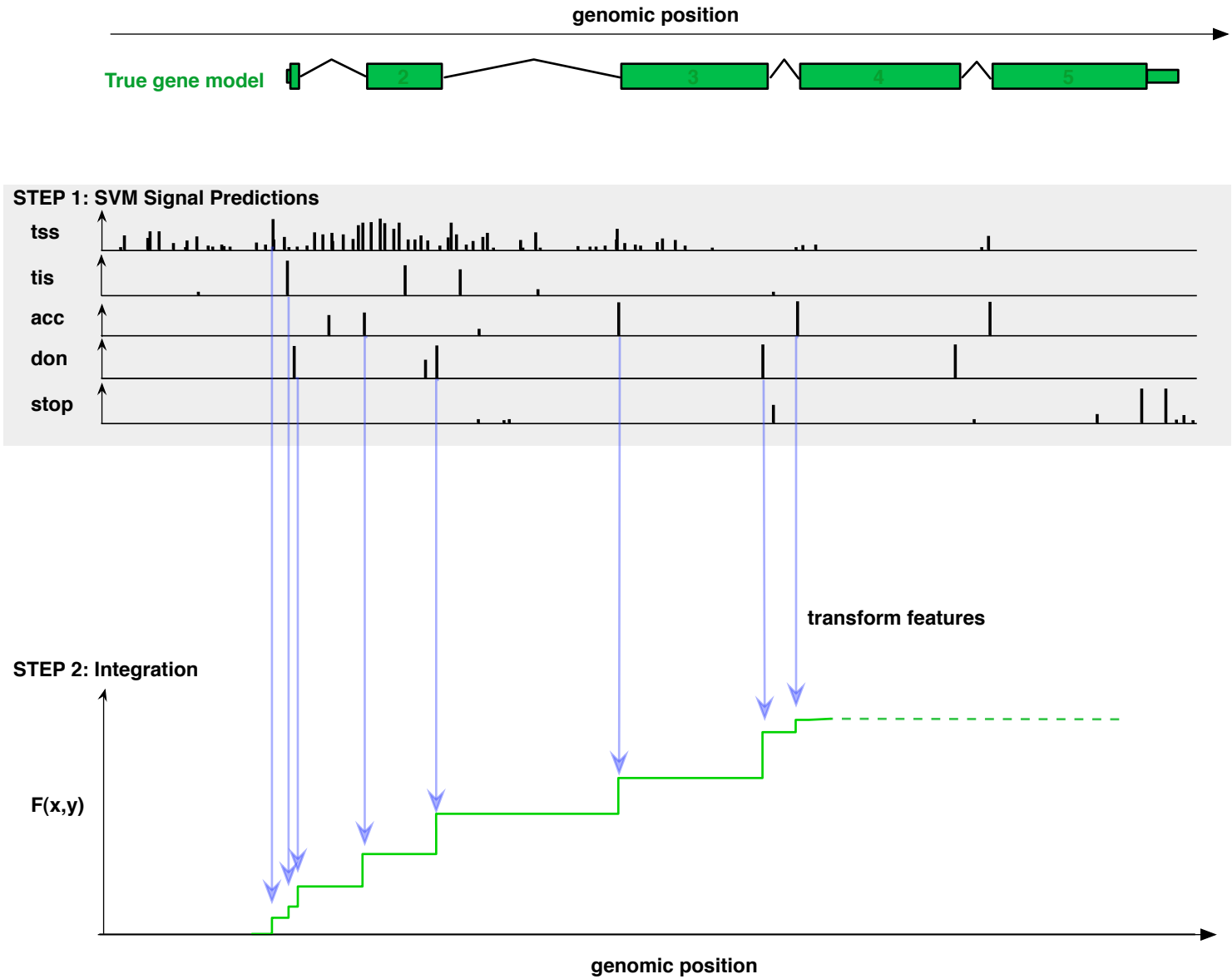
mGene-based Transcript Prediction I



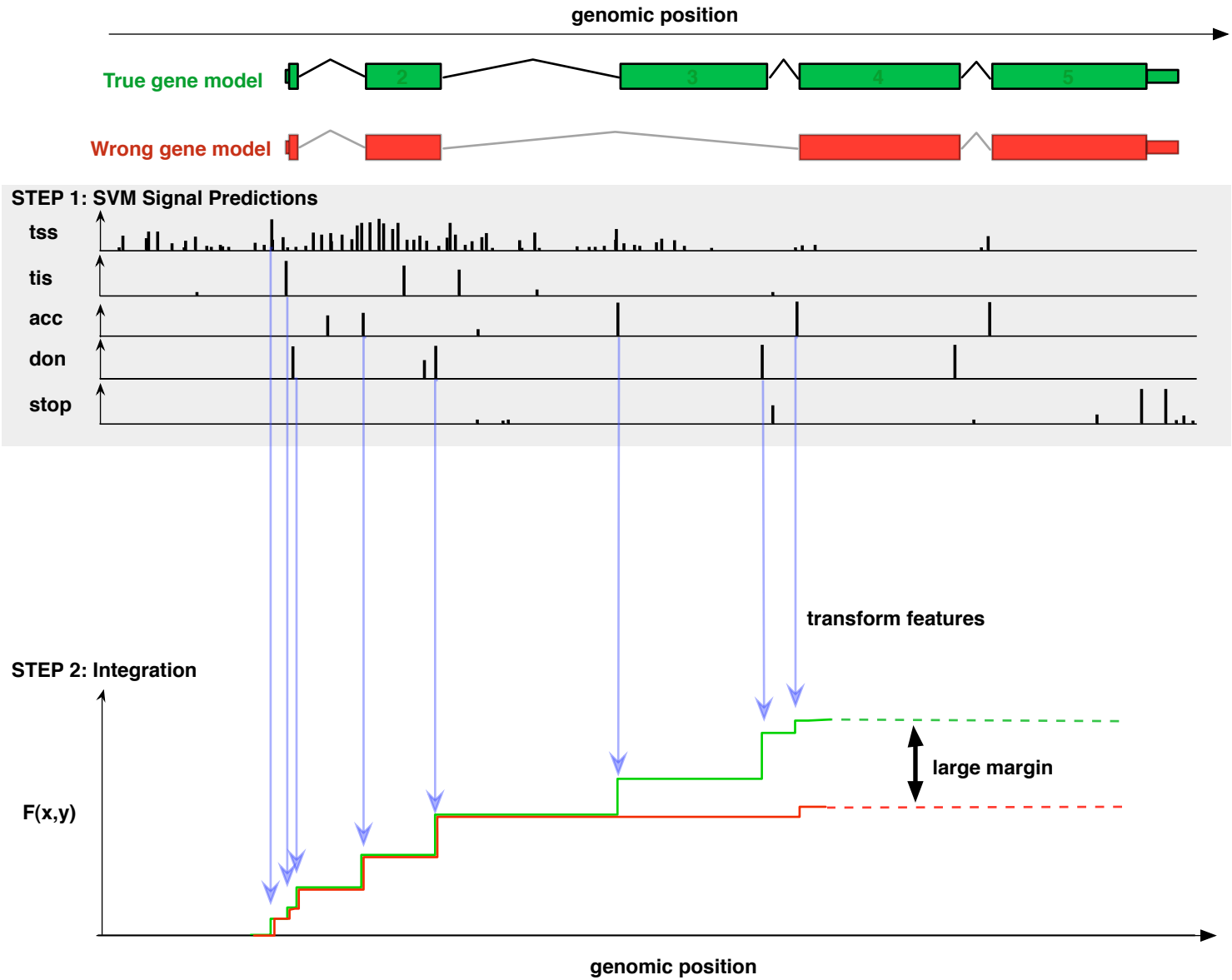




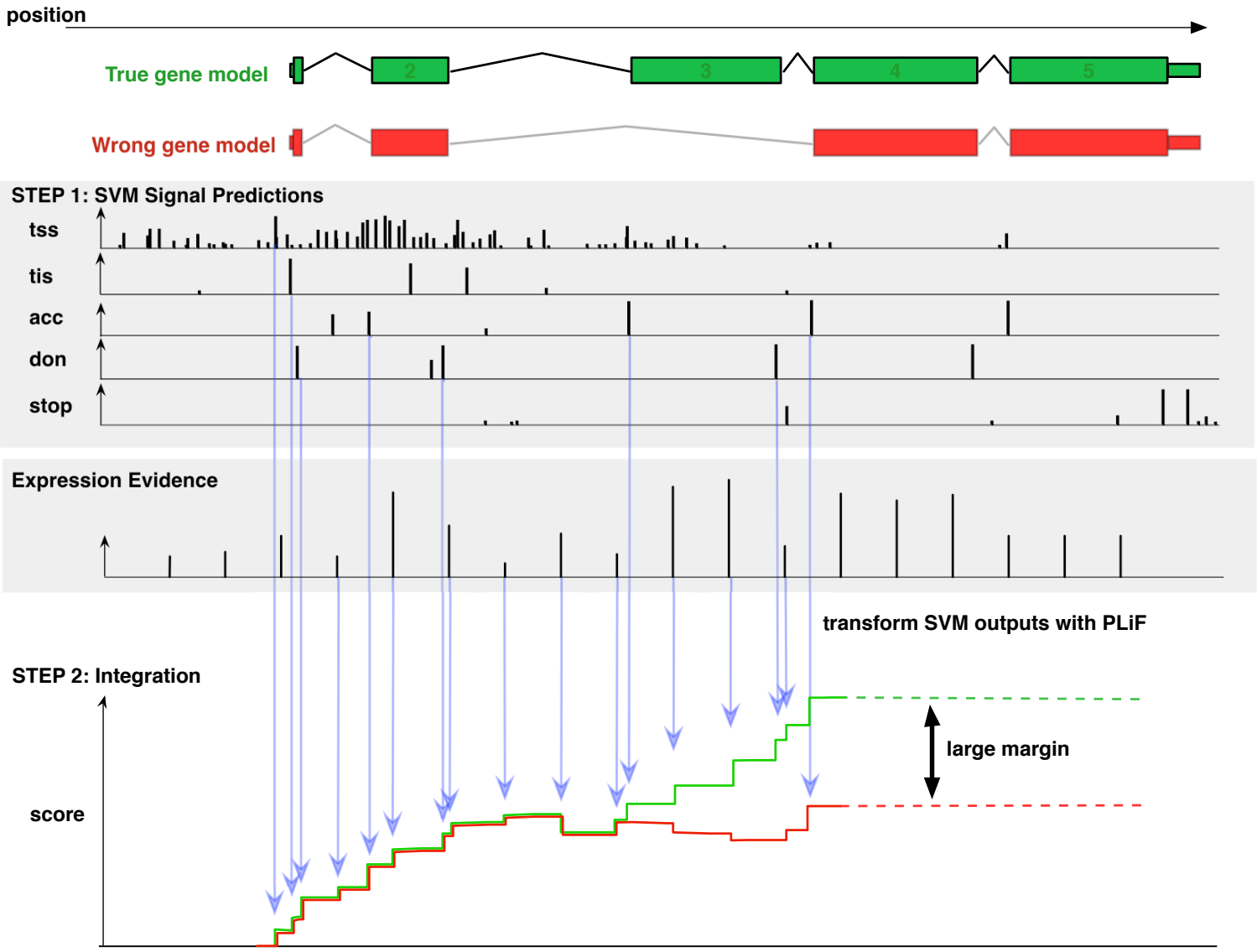
mGene-based Transcript Prediction I



mGene-based Transcript Prediction I



mGene-based Transcript Prediction I



Results using mGene



- Most accurate *ab initio* method in the nGASP genome annotation challenge for *C. elegans* [Coghlan et al., 2008]
- Validation of gene predictions for *C. elegans*: [Schweikert et al., 2009b]

	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: [Schweikert et al., 2009b]

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 [Schweikert et al., 2009a]

Results using mGene

- Most accurate *ab initio* method in the nGASP genome annotation challenge for *C. elegans* [Coghlan et al., 2008]
- Validation of gene predictions for *C. elegans*: [Schweikert et al., 2009b]

	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: [Schweikert et al., 2009b]

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 [Schweikert et al., 2009a]

Results using mGene

- Most accurate *ab initio* method in the nGASP genome annotation challenge for *C. elegans* [Coghlan et al., 2008]
- Validation of gene predictions for *C. elegans*: [Schweikert et al., 2009b]

	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: [Schweikert et al., 2009b]

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 [Schweikert et al., 2009a]

Results using mGene

- Most accurate *ab initio* method in the nGASP genome annotation challenge for *C. elegans* [Coghlan et al., 2008]
- Validation of gene predictions for *C. elegans*: [Schweikert et al., 2009b]

	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: [Schweikert et al., 2009b]

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 [Schweikert et al., 2009a]

Limitations/Extensions



- Gene finding (by far) still not perfect
  - New *ab initio* techniques more accurate
  - Many genes mis-predicted (*C. elegans*: $\approx 50\%$; human: $\approx 80\%$)
  - What is missing?
    - Not enough examples for training?
    - Model complexity?
    - Other information?
- Transcriptome is the result of many complex processes
  - Current methods also ignore other important information:
    - Chromatin structure and methylation patterns
    - RNA structure and processing regulation ...
  - *Ab initio* methods cannot predict alternative transcripts

Limitations/Extensions



- Gene finding (by far) still not perfect
  - New *ab initio* techniques more accurate
  - Many genes mis-predicted (*C. elegans*: $\approx 50\%$; human: $\approx 80\%$)
  - What is missing?
    - Not enough examples for training?
    - Model complexity?
    - Other information?
- Transcriptome is the result of many complex processes
  - Current methods also ignore other important information:
    - Chromatin structure and methylation patterns
    - RNA structure and processing regulation ...
  - *Ab initio* methods cannot predict alternative transcripts

Limitations/Extensions

- Gene finding (by far) still not perfect
  - New *ab initio* techniques more accurate
  - Many genes mis-predicted (*C. elegans*: $\approx 50\%$; human: $\approx 80\%$)
  - What is missing?
    - Not enough examples for training?
    - Model complexity?
    - Other information?
- Transcriptome is the result of many complex processes
  - Current methods also ignore other important information:
    - Chromatin structure and methylation patterns
    - RNA structure and processing regulation ...
  - *Ab initio* methods cannot predict alternative transcripts

Limitations/Extensions



- Gene finding (by far) still not perfect
  - New *ab initio* techniques more accurate
  - Many genes mis-predicted (*C. elegans*: $\approx 50\%$; human: $\approx 80\%$)
  - What is missing?
    - Not enough examples for training?
    - Model complexity?
    - Other information?
- Transcriptome is the result of many complex processes
  - Current methods also ignore other important information:
    - Chromatin structure and methylation patterns
    - RNA structure and processing regulation ...
  - *Ab initio* methods cannot predict alternative transcripts

Gene Finding + Auxiliary Measurements

[Behr et al., 2009, in prep.]



Is there other information that may be used by cellular processes that can improve prediction results?

Preliminary study: (*A. thaliana*) transcript level $(SN + SP)/2$

1. mGene (*ab initio*) . . . **73.3%**
2. . . . with DNA methylation (1 tissue) **76.1%**
3. . . . with Nucleosome position predictions **78.0%**
4. . . . with RNA secondary structure predictions **76.7%**

In progress:

- Study effect of other information sources for human gene prediction
- Ideally, consider measurements from the same sample

Gene Finding + Auxiliary Measurements

[Behr et al., 2009, in prep.]



Is there other information that may be used by cellular processes that can improve prediction results?

Preliminary study: (*A. thaliana*) transcript level $(SN + SP)/2$

1. mGene (*ab initio*) . . . **73.3%**
2. . . . with DNA methylation (1 tissue) **76.1%**
3. . . . with Nucleosome position predictions **78.0%**
4. . . . with RNA secondary structure predictions **76.7%**

In progress:

- Study effect of other information sources for human gene prediction
- Ideally, consider measurements from the same sample

Gene Finding + Auxiliary Measurements

[Behr et al., 2009, in prep.]



Is there other information that may be used by cellular processes that can improve prediction results?

Preliminary study: (*A. thaliana*) transcript level $(SN + SP)/2$

1. mGene (*ab initio*) . . . **73.3%**
2. . . . with DNA methylation (1 tissue) **76.1%**
3. . . . with Nucleosome position predictions **78.0%**
4. . . . with RNA secondary structure predictions **76.7%**

In progress:

- Study effect of other information sources for human gene prediction
- Ideally, consider measurements from the same sample

Gene Finding + Auxiliary Measurements

[Behr et al., 2009, in prep.]



Is there other information that may be used by cellular processes that can improve prediction results?

Preliminary study: (*A. thaliana*) transcript level $(SN + SP)/2$

1. mGene (*ab initio*) . . . **73.3%**
2. . . . with DNA methylation (1 tissue) **76.1%**
3. . . . with Nucleosome position predictions **78.0%**
4. . . . with RNA secondary structure predictions **76.7%**

In progress:

- Study effect of other information sources for human gene prediction
- Ideally, consider measurements from the same sample

Gene Finding + Auxiliary Measurements

[Behr et al., 2009, in prep.]

Is there other information that may be used by cellular processes that can improve prediction results?

Preliminary study: (*A. thaliana*) transcript level $(SN + SP)/2$

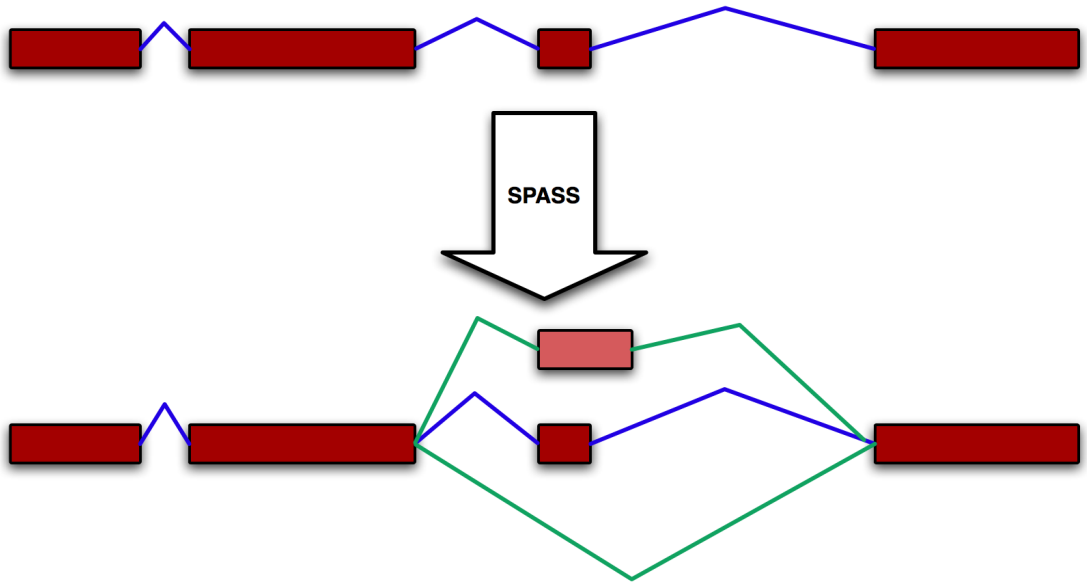
1. mGene (*ab initio*) . . . **73.3%**
2. . . . with DNA methylation (1 tissue) **76.1%**
3. . . . with Nucleosome position predictions **78.0%**
4. . . . with RNA secondary structure predictions **76.7%**

In progress:

- Study effect of other information sources for human gene prediction
- Ideally, consider measurements from the same sample

Alternative Transcripts: First Steps

Predictions of alternative splicing



- Predict novel alternative splicing as independent events
- Only use sequence information [Rätsch et al., 2005]
- Requires known gene structures: Use *ab initio* predictions

Alternative Transcripts: More Steps



- Combine gene finding with prediction of alternative splicing
  - Machine learning challenge:

    Input: DNA sequence
    Output: Splicing graph

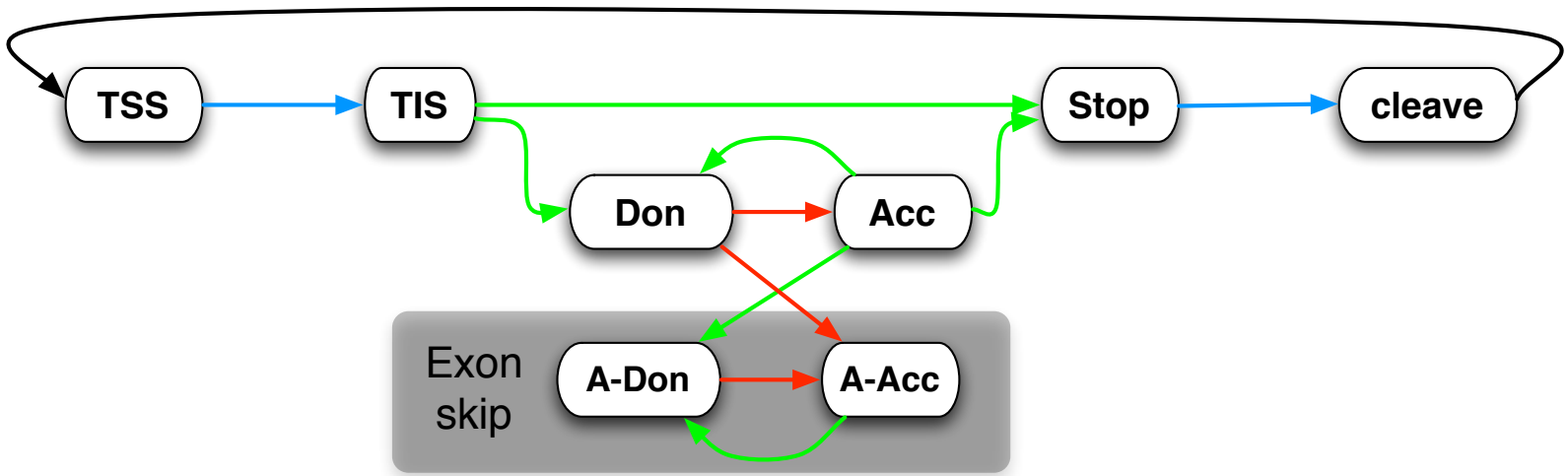
- mGene approach can be extended to
  - Predict **"simple splicing graphs"**

- But, without **clean data** we cannot train the model ...

Alternative Transcripts: More Steps

- Combine gene finding with prediction of alternative splicing
  - Machine learning challenge:

    Input: DNA sequence
    Output: Splicing graph
- mGene approach can be extended to
  - Predict **"simple splicing graphs"**

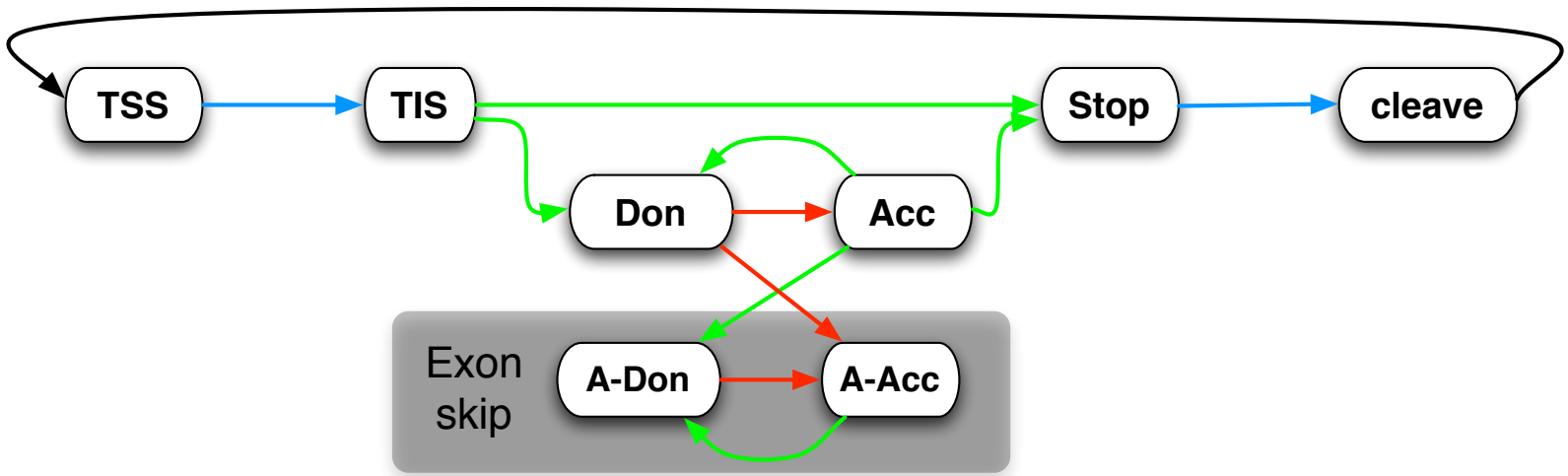


- But, without **clean data** we cannot train the model ...

Alternative Transcripts: More Steps

- Combine gene finding with prediction of alternative splicing
  - Machine learning challenge:

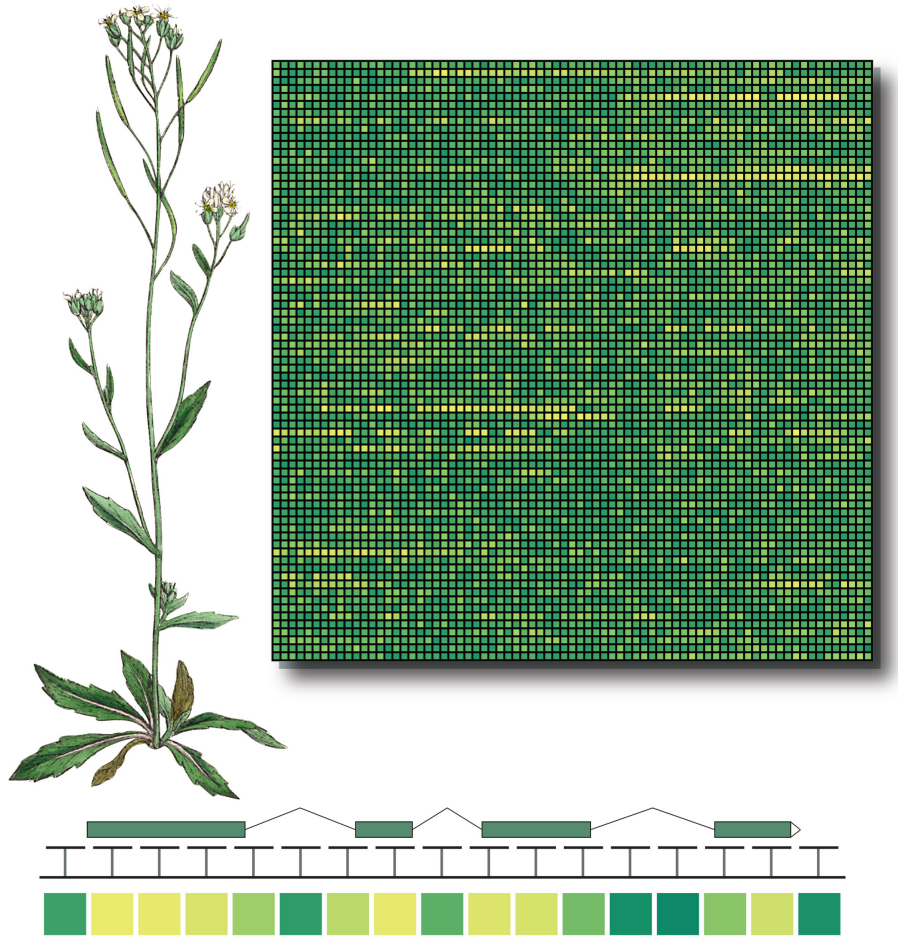
    Input: DNA sequence
    Output: Splicing graph
- mGene approach can be extended to
  - Predict **"simple splicing graphs"**



- But, without **clean data** we cannot train the model ...

Studying Transcriptome Plasticity with Tiling Arrays

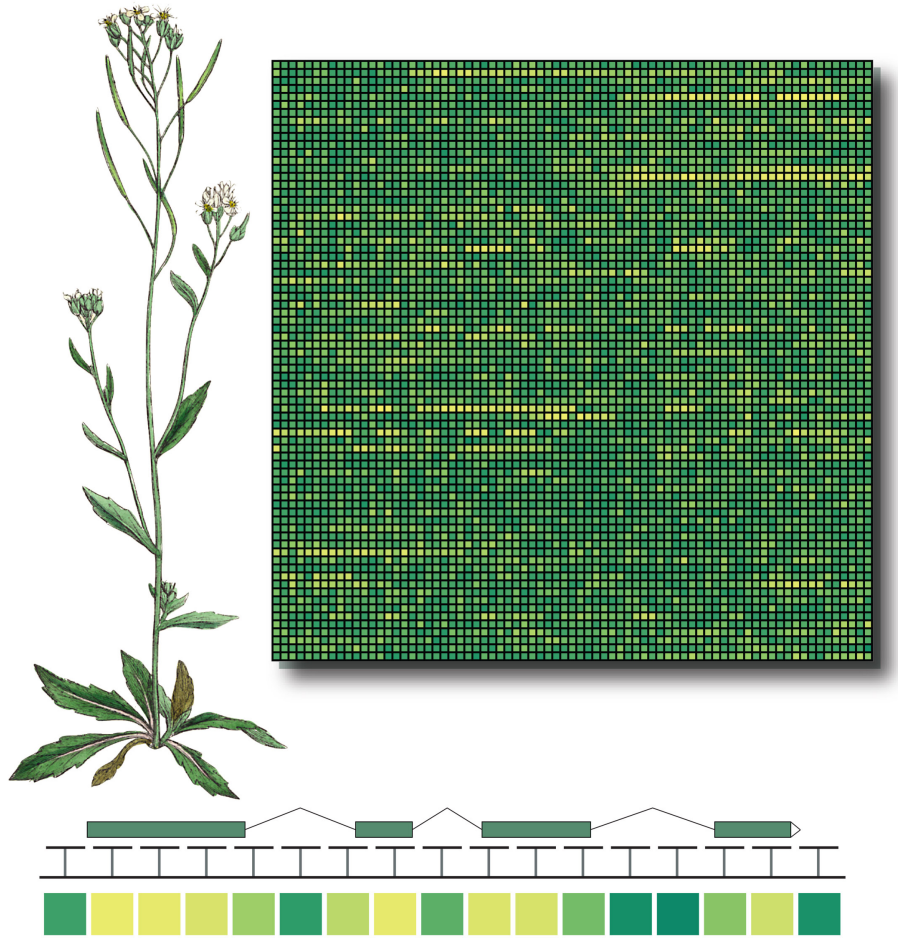
- Identify the full complement of genes in the model plant *Arabidopsis thaliana*
- Profile transcriptome changes during development and under stress
- Characterize global effects of sRNA processing factors on the transcriptome



Studying Transcriptome Plasticity with Tiling Arrays



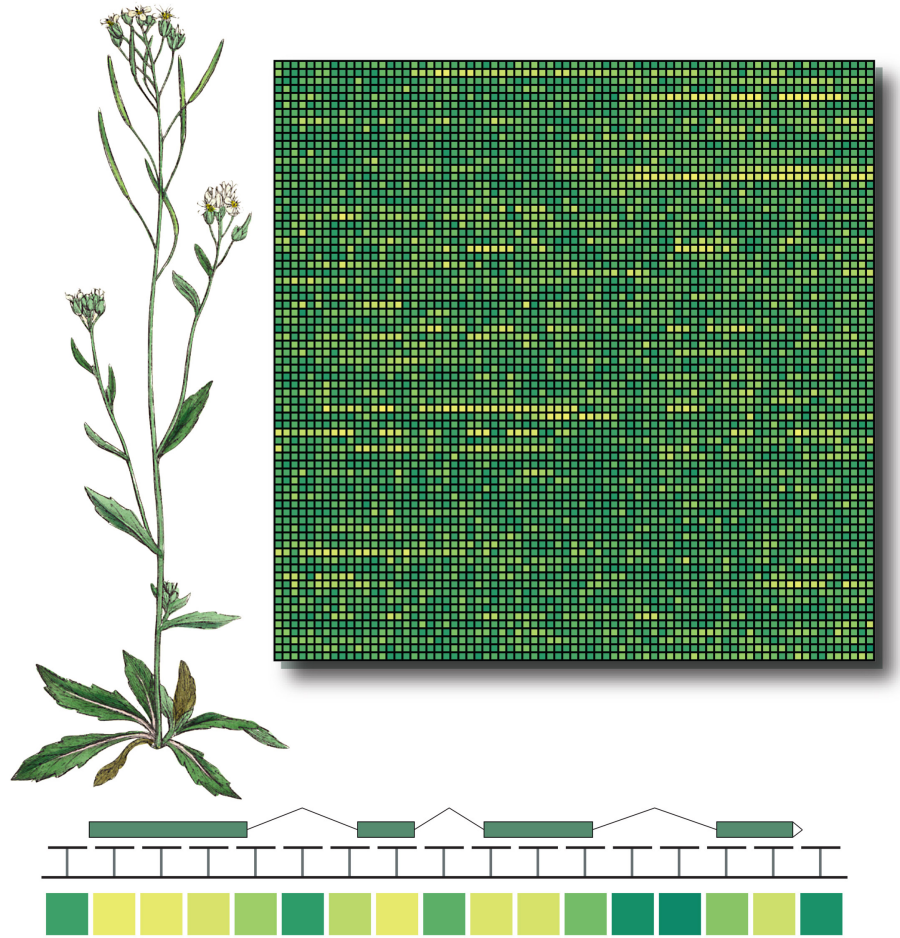
- Identify the full complement of genes in the model plant *Arabidopsis thaliana*
- Profile transcriptome changes during development and under stress
- Characterize global effects of sRNA processing factors on the transcriptome



Studying Transcriptome Plasticity with Tiling Arrays



- Identify the full complement of genes in the model plant *Arabidopsis thaliana*
- Profile transcriptome changes during development and under stress
- Characterize global effects of sRNA processing factors on the transcriptome



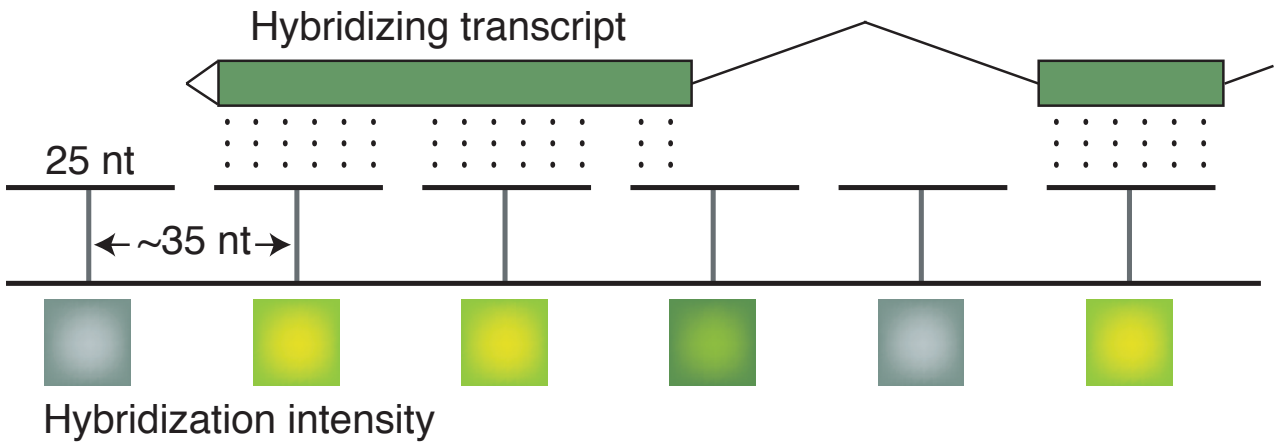
Tiling Arrays for Transcriptome Studies



Whole-genome DNA microarray

- Whole-genome, unbiased, quantitative expression measurements
- Cost-effective → transcriptome profiling under various conditions
- Noisy measurements → challenging data analysis

Tiling Arrays for Transcriptome Studies



- Whole-genome, unbiased, quantitative expression measurements
- Cost-effective $\longrightarrow$ transcriptome profiling under various conditions
- Noisy measurements $\longrightarrow$ challenging data analysis

Normalization Pipeline



- Background normalization
- Quantile normalization → comparisons between arrays
- Transcript normalization → probe sequence bias reduced

Key idea: Learn deviation between observed signal intensities and ideal, constant transcript intensity from the probe sequence using an array of linear regression models

G. Zeller et al., Pac. Symp. Biocomp. (2008)

Normalization Pipeline



- Background normalization
- Quantile normalization → comparisons between arrays
- Transcript normalization → probe sequence bias reduced

Key idea: Learn deviation between observed signal intensities and ideal, constant transcript intensity from the probe sequences using an array of linear regression models

G. Zeller et al., Pac. Symp. Biocomp. (2008)

Normalization Pipeline



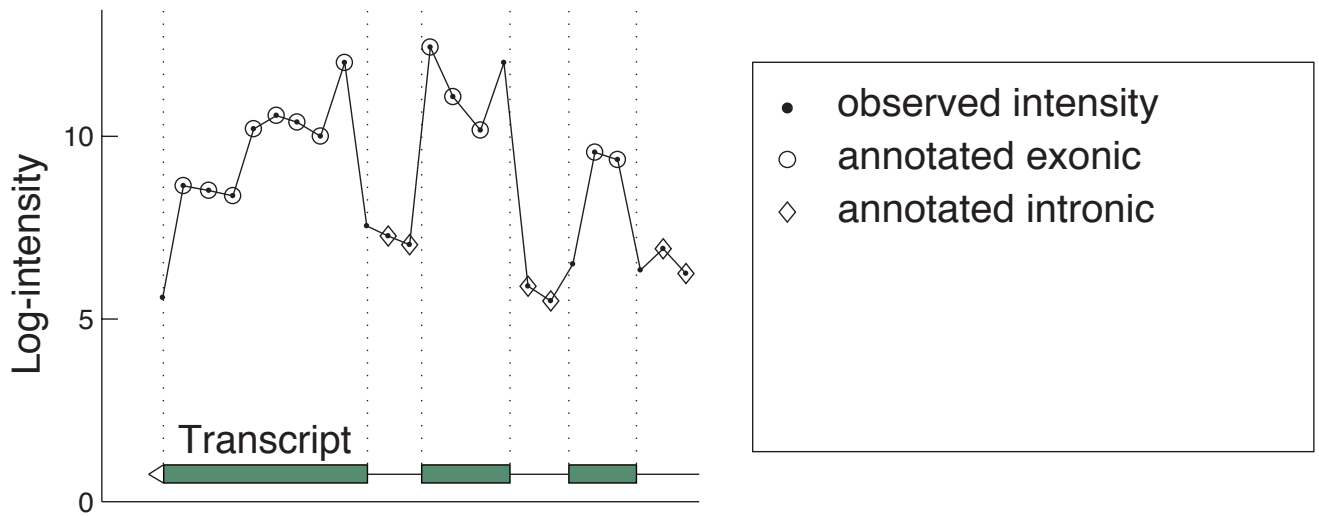
- Background normalization
- Quantile normalization $\longrightarrow$ comparisons between arrays
- Transcript normalization $\longrightarrow$ probe sequence bias reduced

Key idea: Learn deviation between observed signal intensities and ideal, constant transcript intensity from tiling probe sequences using an array of linear regression models

G. Zeller et al., Pac. Symp. Biocomp. (2008)

Normalization Pipeline

- Background normalization
- Quantile normalization → comparisons between arrays
- Transcript normalization → probe sequence bias reduced

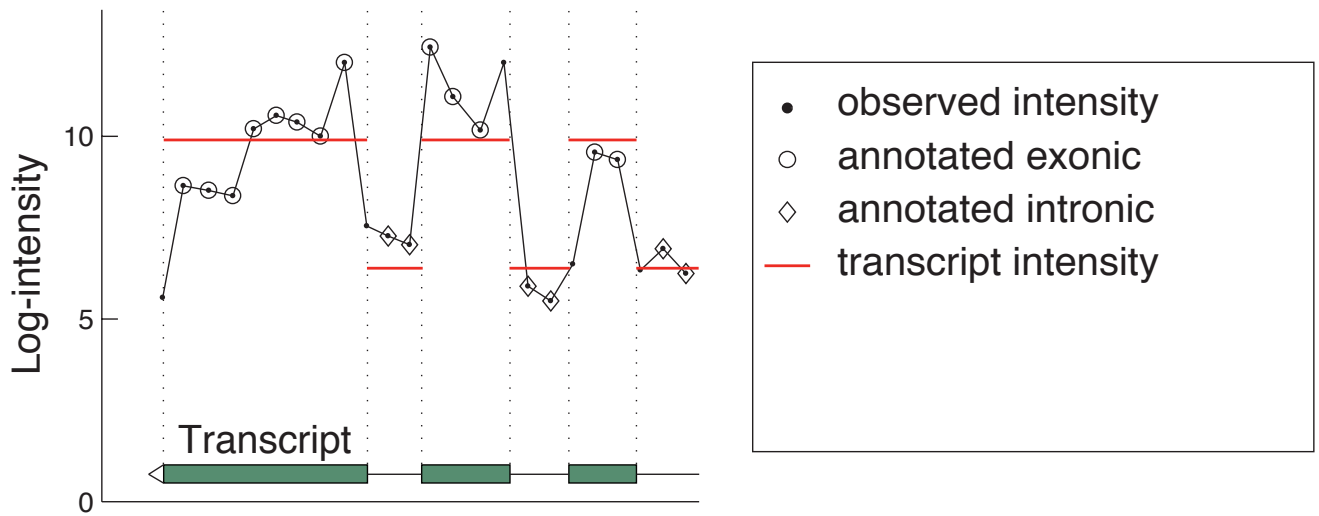


Key idea: Learn deviation between observed signal intensities and ideal, constant transcript intensity from tiling probe sequences using an array of linear regression models

G. Zeller et al., Pac. Symp. Biocomp. (2008)

Normalization Pipeline

- Background normalization
- Quantile normalization → comparisons between arrays
- Transcript normalization → probe sequence bias reduced

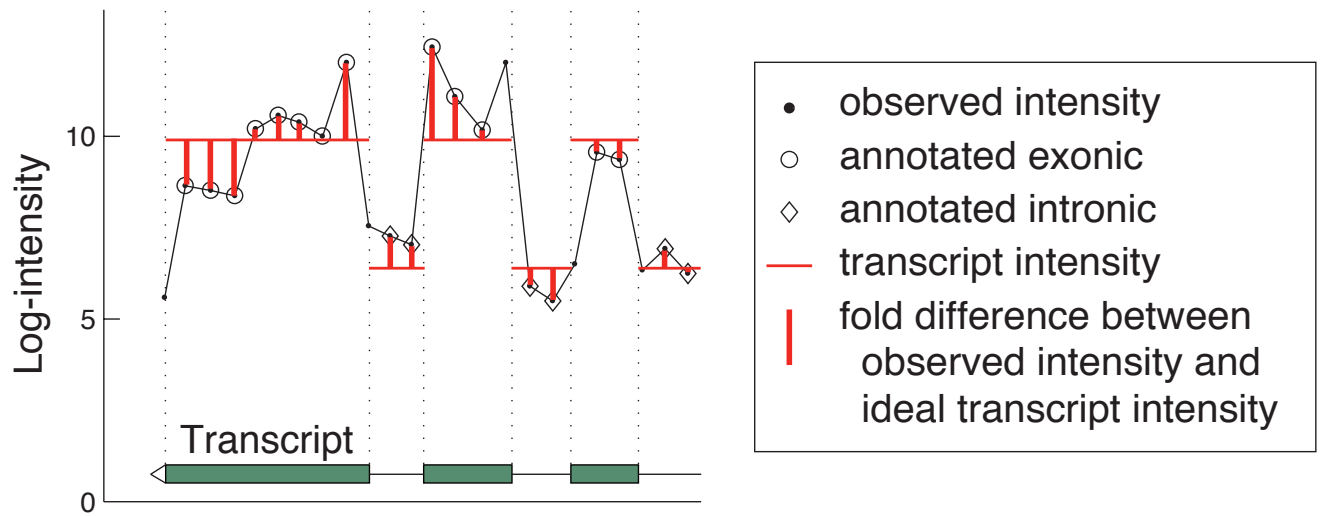


Key idea: Learn deviation between observed signal intensities and ideal, constant transcript intensity from tiling probe sequences using an array of linear regression models

G. Zeller et al., Pac. Symp. Biocomp. (2008)

Normalization Pipeline

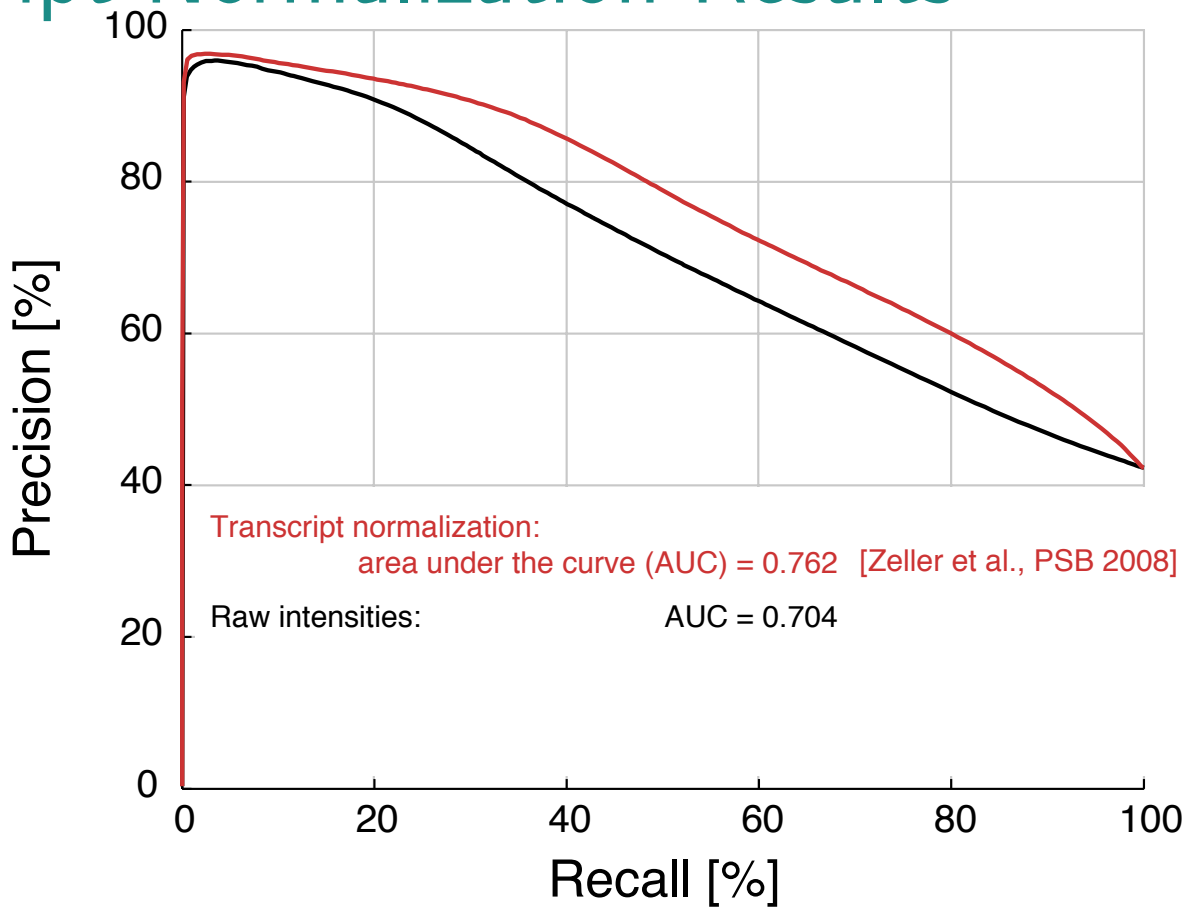
- Background normalization
- Quantile normalization → comparisons between arrays
- Transcript normalization → probe sequence bias reduced



Key idea: Learn deviation between observed signal intensities and ideal, constant transcript intensity from tiling probe sequences using an array of linear regression models

G. Zeller et al., Pac. Symp. Biocomp. (2008)

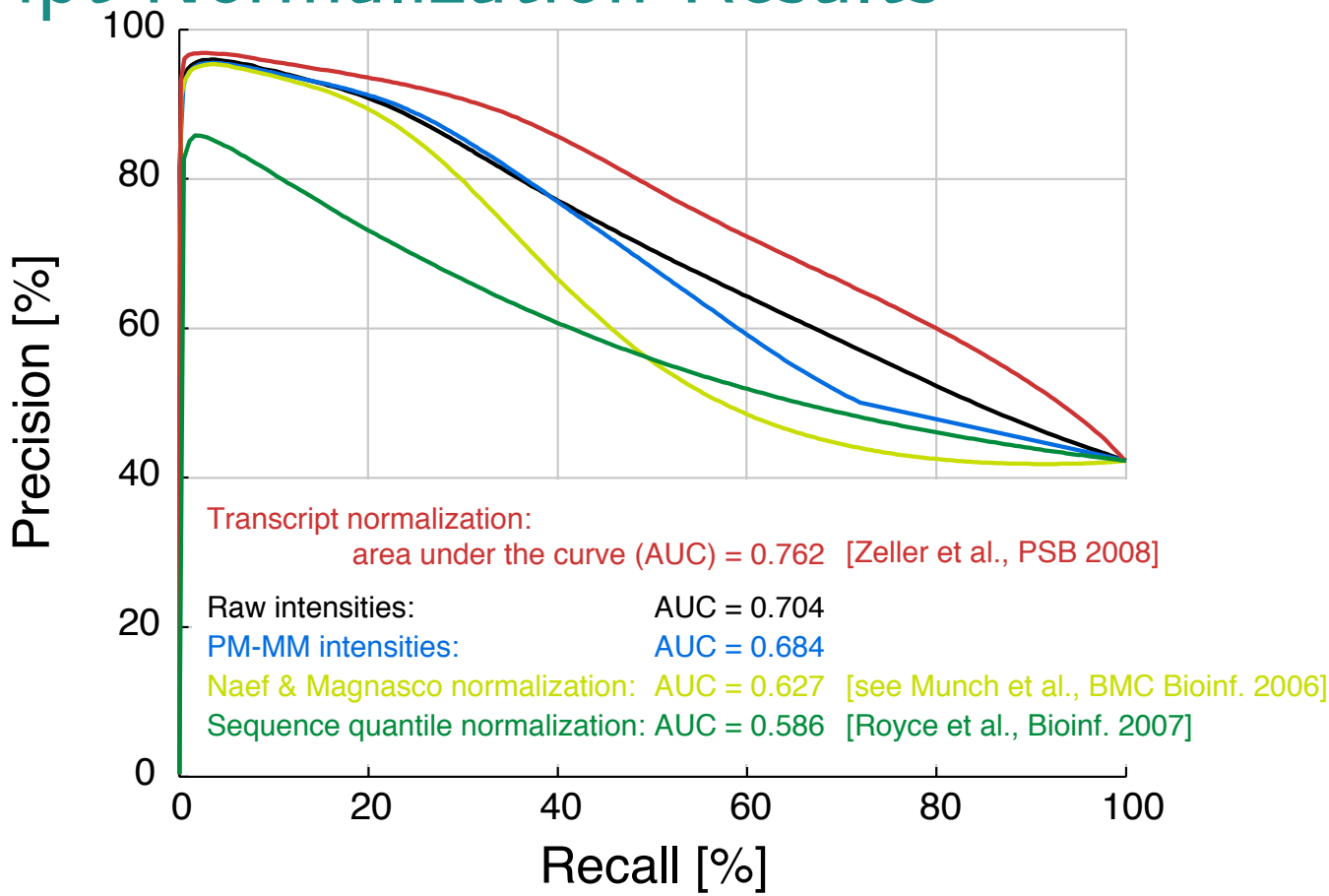
Transcript Normalization Results



Transcript normalization (TN) results in better separation between background and exon signals → improves transcript identification

`www.fml.mpg.de/raetsch/research/tiling/`

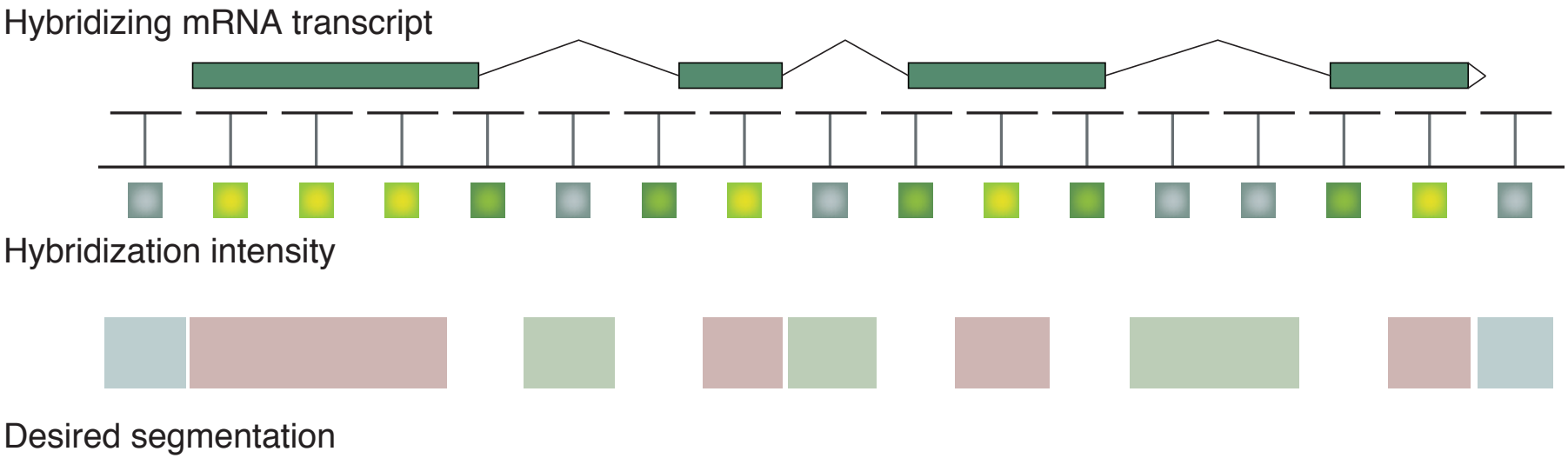
Transcript Normalization Results



Transcript normalization (TN) results in better separation between background and exon signals → improves transcript identification

`www.fml.mpg.de/raetsch/research/tiling/`

Tiling Array Segmentation

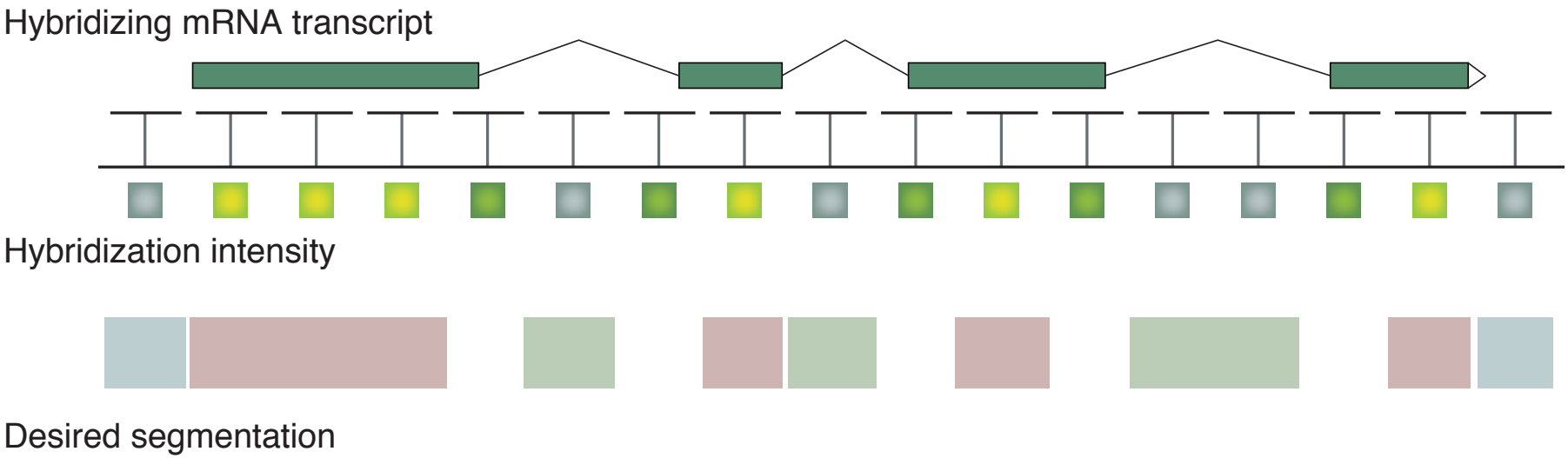


Goal: Characterize tiling probes either as intergenic, exonic or intronic

margin-based segmentation of tiling array data (mSTAD)

`www.fml.mpg.de/raetsch/research/tiling/`

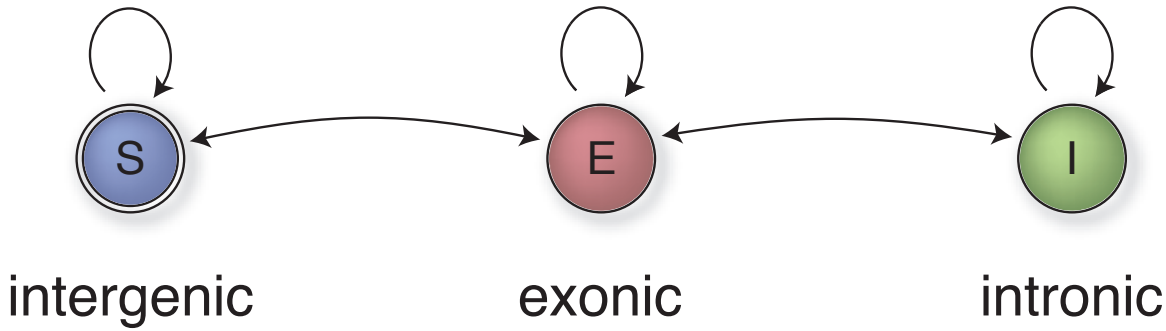
Tiling Array Segmentation



Goal: Characterize tiling probes either as intergenic, exonic or intronic margin-based segmentation of tiling array data (mSTAD)

`www.fml.mpg.de/raetsch/research/tiling/`

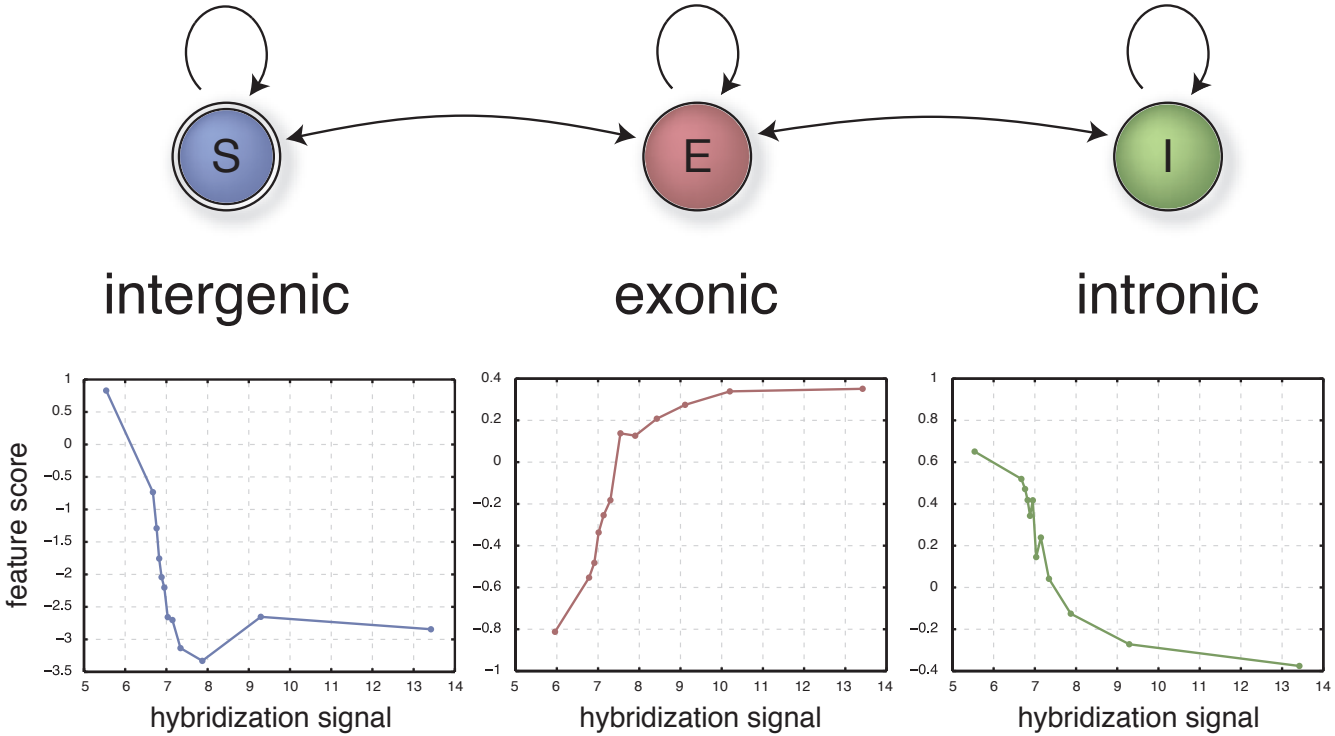
The mSTAD Approach



- Learn to associate a state with each probe given its hybridization signal and local context
- HM-SVM training: Optimize transformations: signal → score

G. Zeller et al., Pac. Symp. Biocomp. (2008)

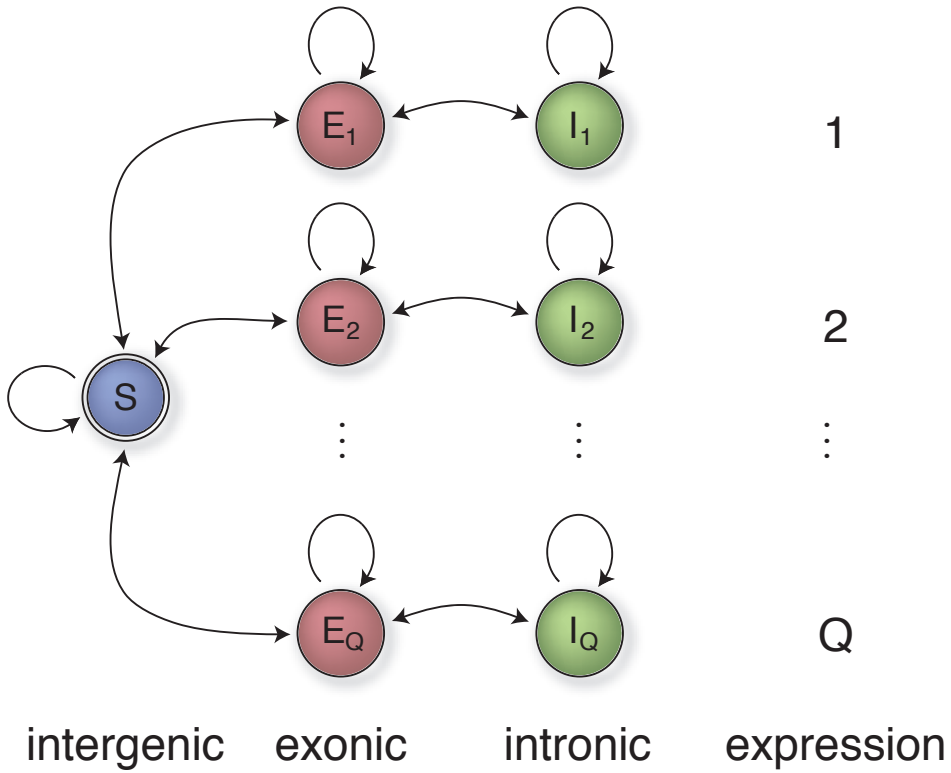
The mSTAD Approach



- Learn to associate a state with each probe given its hybridization signal and local context
- HM-SVM training: Optimize transformations: signal $\longrightarrow$ score

G. Zeller et al., Pac. Symp. Biocomp. (2008)

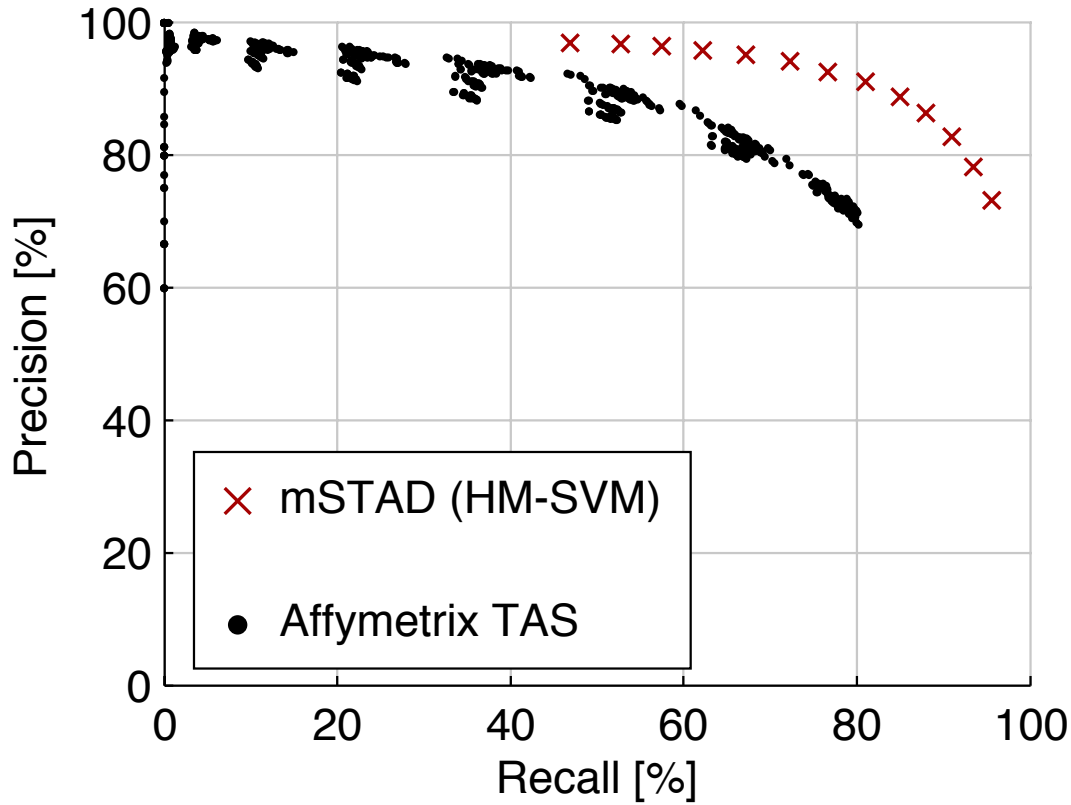
The mSTAD Approach



- Learn to associate a state with each probe given its hybridization signal and local context
- HM-SVM training: Optimize transformations: signal $\longrightarrow$ score

G. Zeller et al., Pac. Symp. Biocomp. (2008)

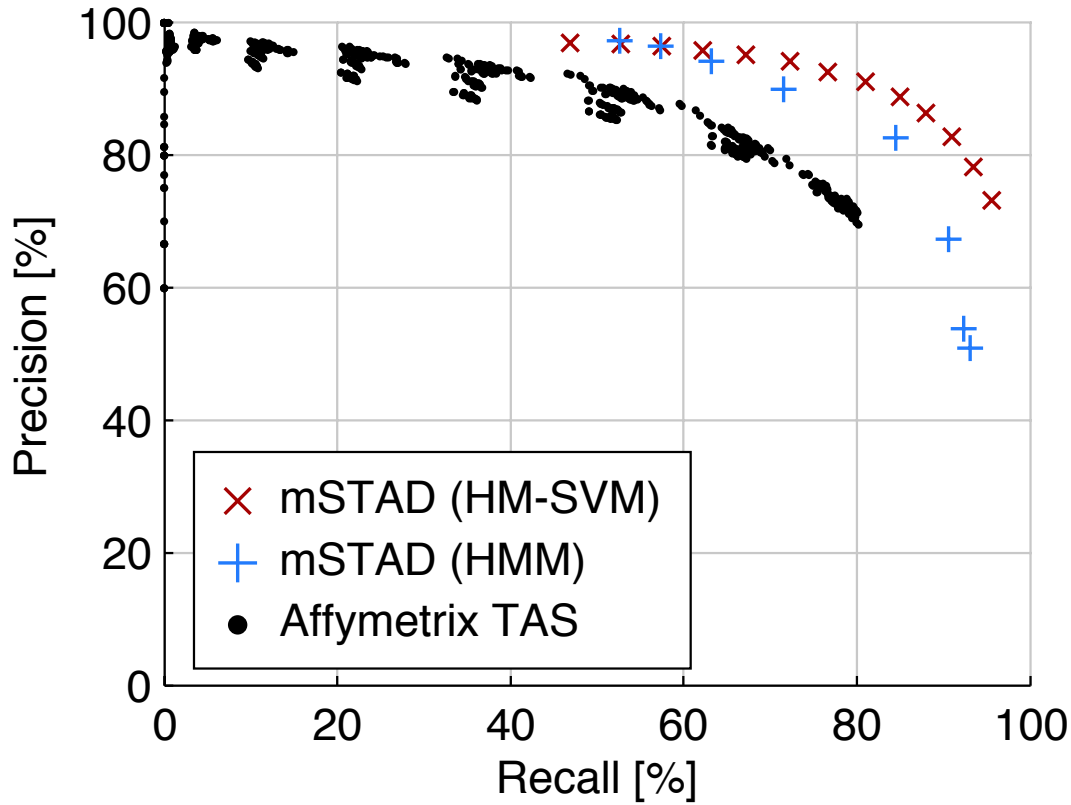
Method Comparison



Substantially improved exon probe recognition over the most widely used "transfrag" method

`www.fml.mpg.de/raetsch/research/tiling/`

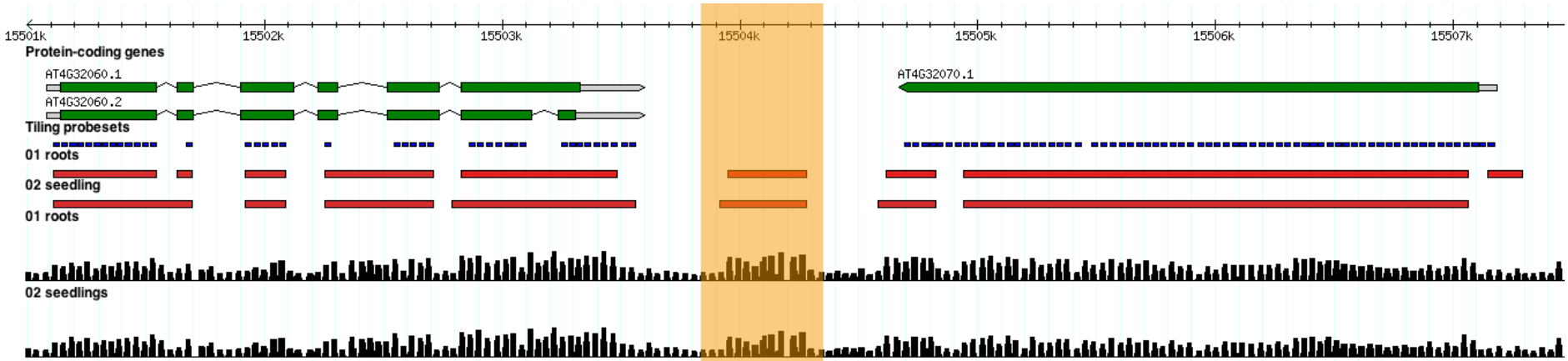
Method Comparison



Substantially improved exon probe recognition over the most widely used "transfrag" method

`www.fml.mpg.de/raetsch/research/tiling/`

Transcriptionally Active Regions (TARs)

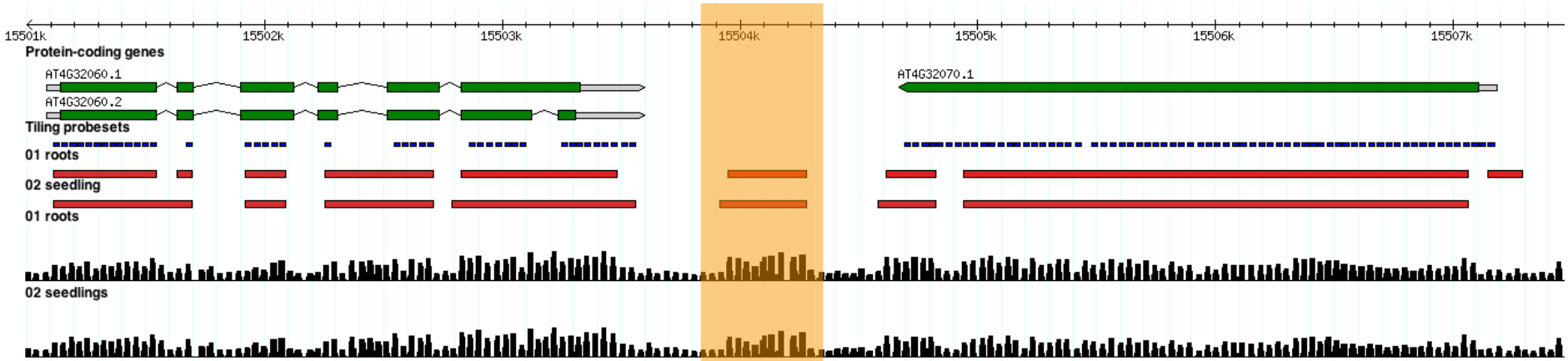


new TAR, not previously apparent from ESTs or cDNAs

www.weigelworld.org/resources/microarray/at-tax/

S. Laubinger et al., Genome Biol. (2008)

Transcriptionally Active Regions (TARs)

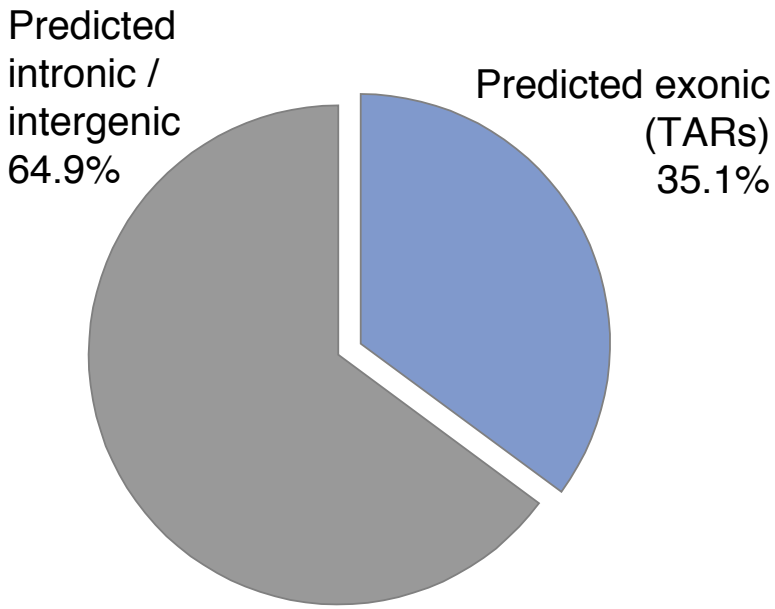


new TAR, not previously apparent from ESTs or cDNAs

`www.weigelworld.org/resources/microarray/at-tax/`

S. Laubinger et al., Genome Biol. (2008)

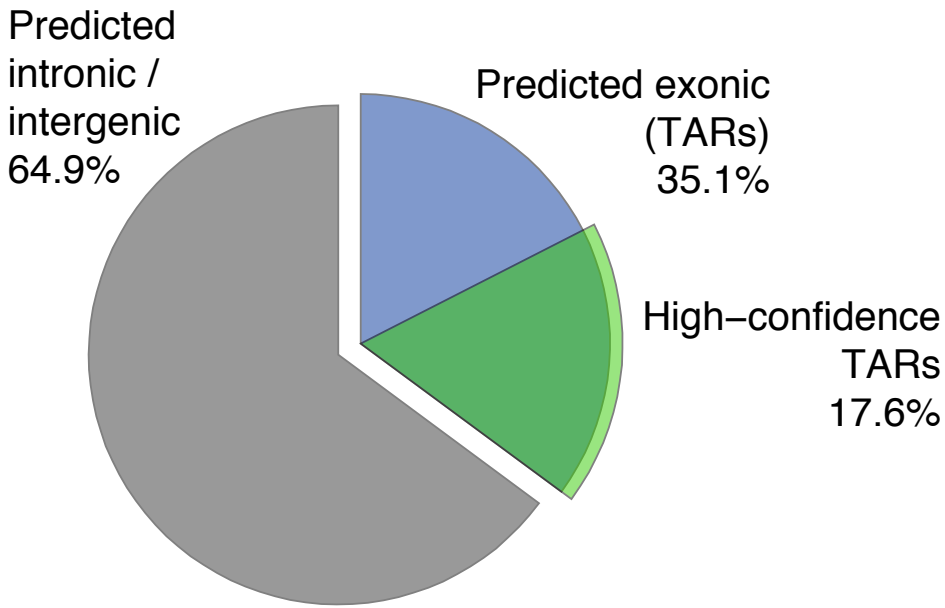
New TARs



- Between 1,107 and 1,947 newly predicted high-confidence TARs per tissue / developmental stage (total length 242 to 406 kb) are absent from the annotation and not covered by ESTs/cDNAs
- 37 of 47 (>75%) RT-PCR validations successful

S. Laubinger et al., Genome Biol. (2008)

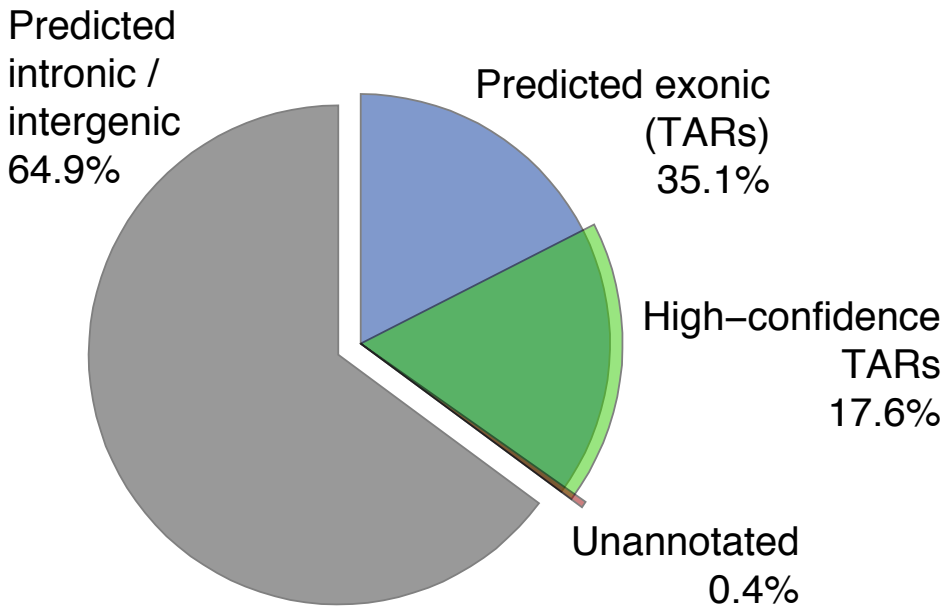
New TARs



- Between 1,107 and 1,947 newly predicted high-confidence TARs per tissue / developmental stage (total length 242 to 406 kb) are absent from the annotation and not covered by ESTs/cDNAs
- 37 of 47 (>75%) RT-PCR validations successful

S. Laubinger et al., Genome Biol. (2008)

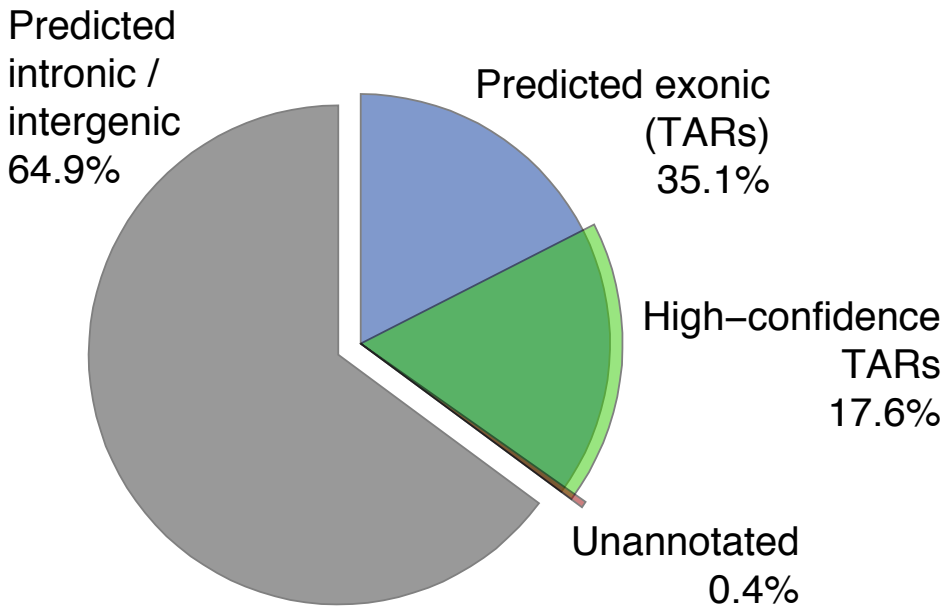
New TARs



- Between 1,107 and 1,947 newly predicted high-confidence TARs per tissue / developmental stage (total length 242 to 406 kb) are absent from the annotation and not covered by ESTs/cDNAs
- 37 of 47 (>75%) RT-PCR validations successful

S. Laubinger et al., Genome Biol. (2008)

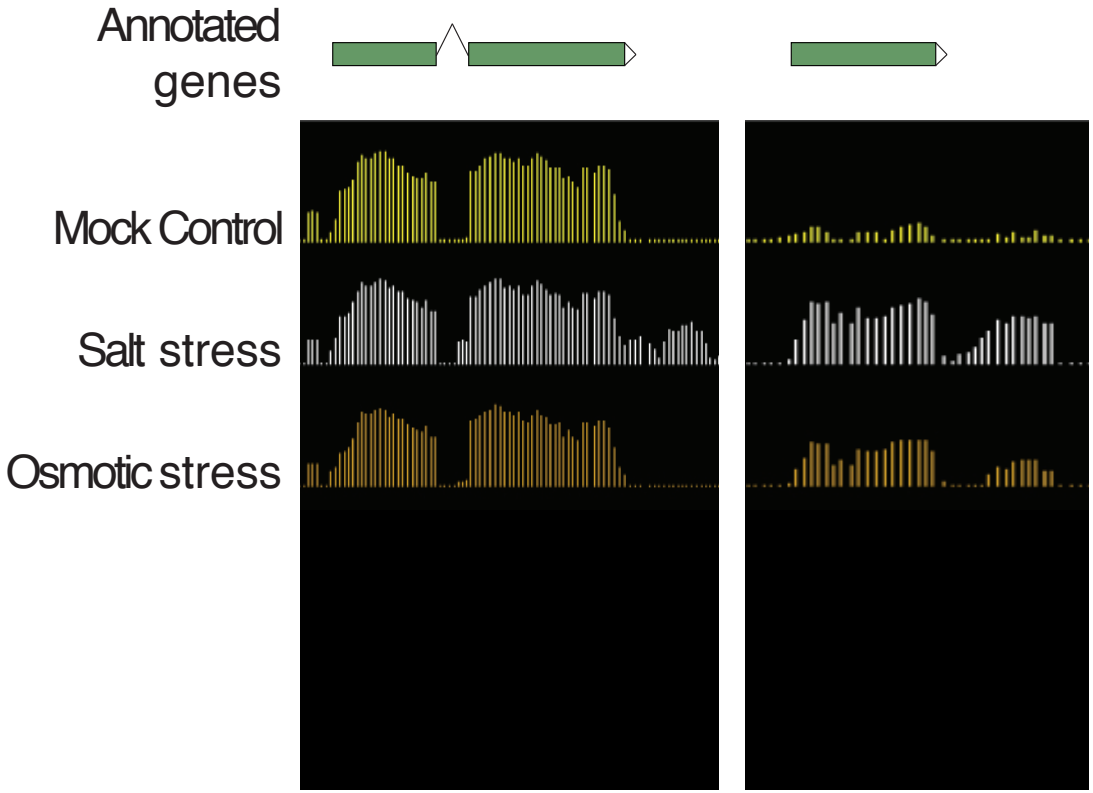
New TARs



- Between 1,107 and 1,947 newly predicted high-confidence TARs per tissue / developmental stage (total length 242 to 406 kb) are absent from the annotation and not covered by ESTs/cDNAs
- 37 of 47 (>75%) RT-PCR validations successful

S. Laubinger et al., Genome Biol. (2008)

New Stress-Induced TARs

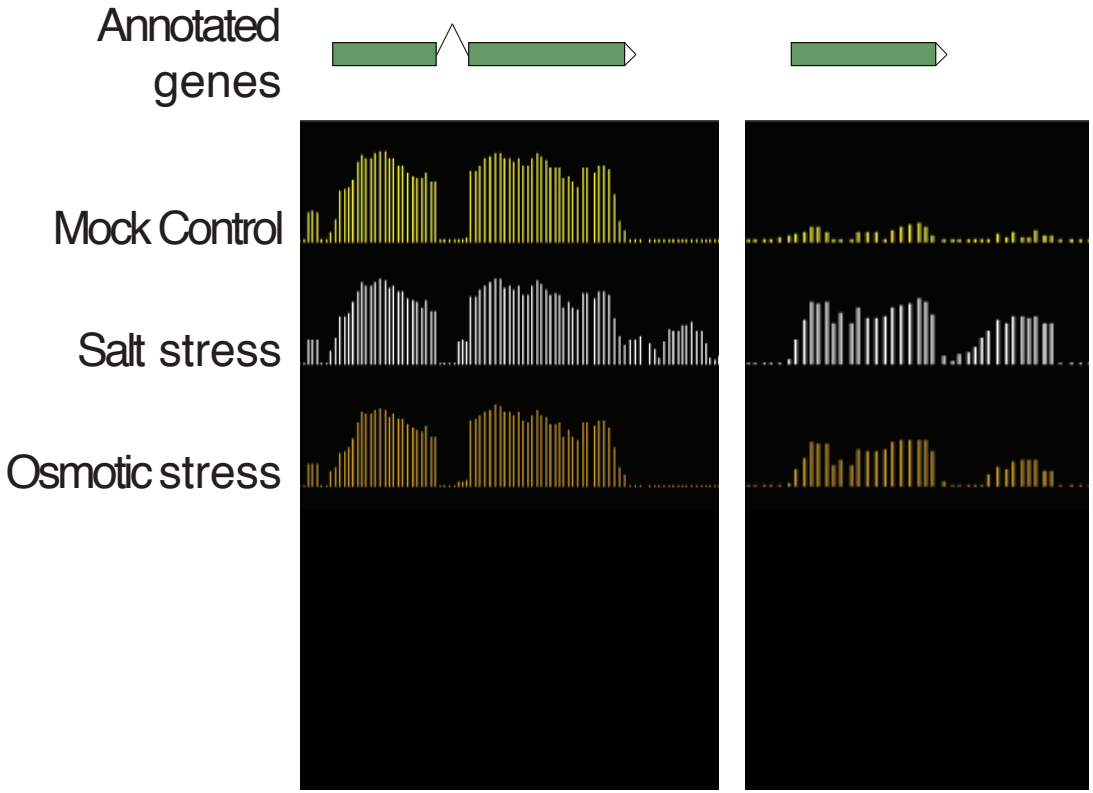


Identify TARs induced under stress (by significant increase of hybridization signal)

- We found candidates for...
  - stress-specific alternative transcripts
  - overlapping (possibly regulatory) transcripts
  - new stress-induced genes...

G. Zeller et al., Plant J. (2009)

New Stress-Induced TARs

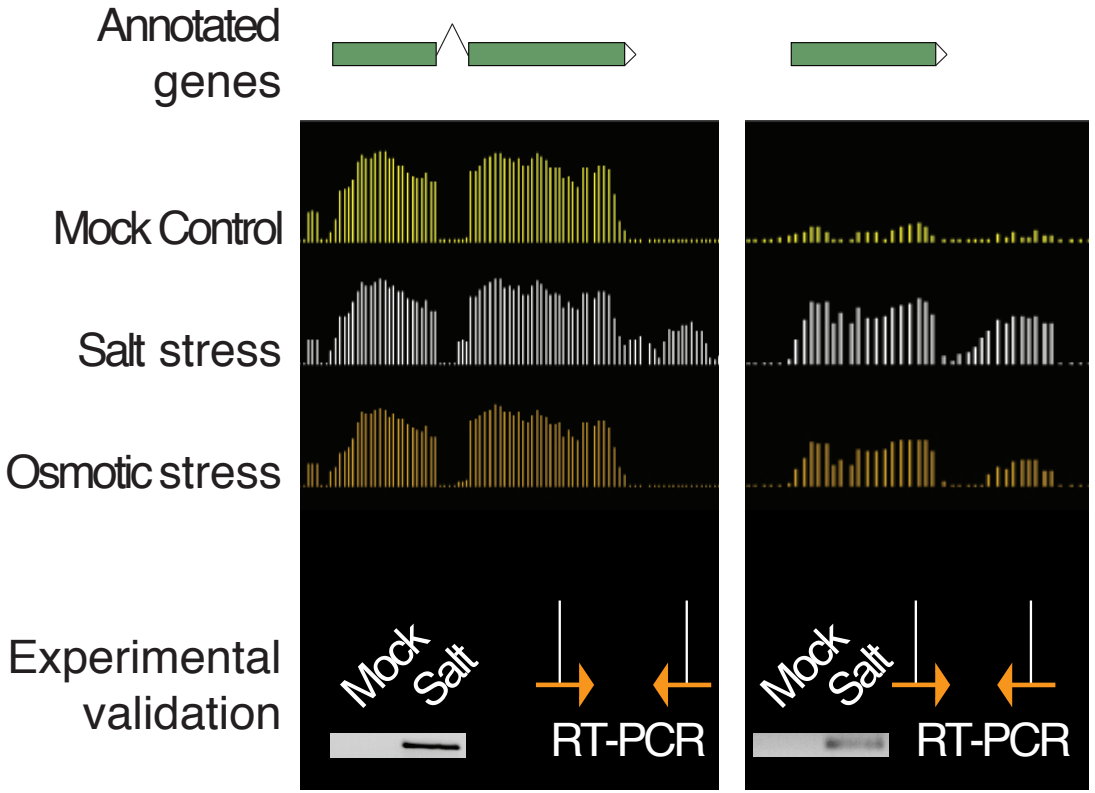


Identify TARs induced under stress (by significant increase of hybridization signal)

- We found candidates for...
  - stress-specific alternative transcripts
  - overlapping (possibly regulatory) transcripts
  - new stress-induced genes...

G. Zeller et al., Plant J. (2009)

New Stress-Induced TARs

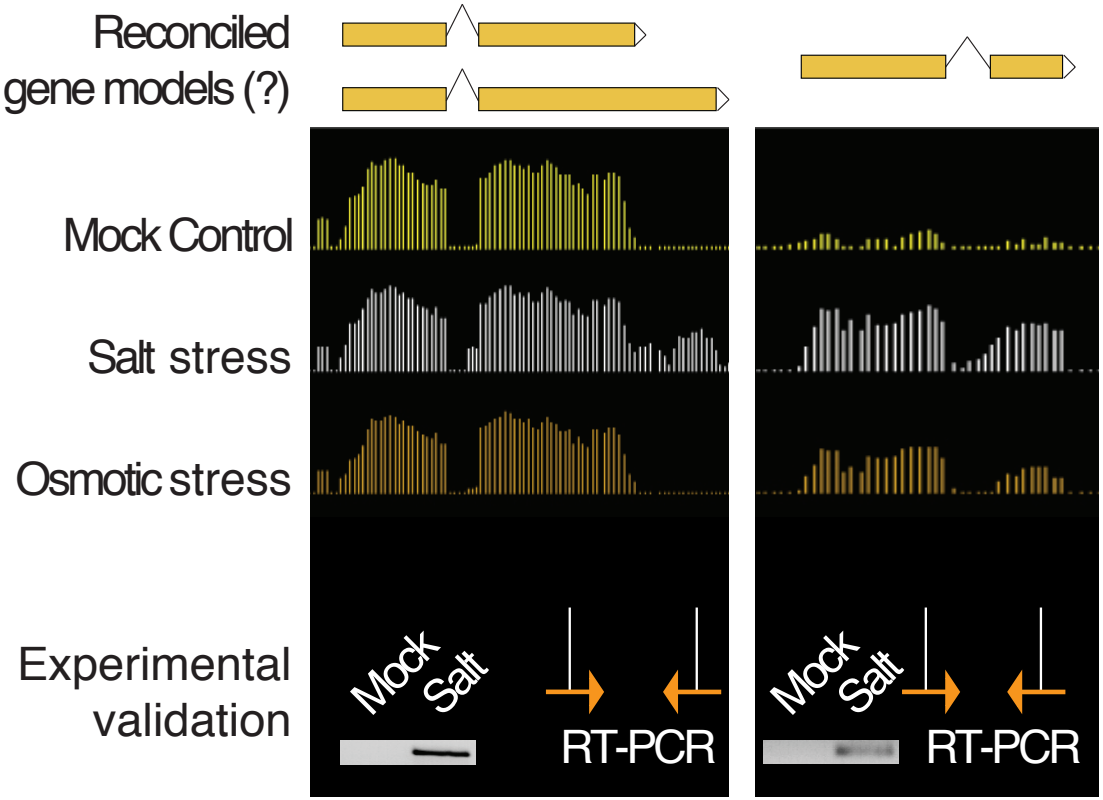


Identify TARs induced under stress (by significant increase of hybridization signal)

- We found candidates for...
  - stress-specific alternative transcripts
  - overlapping (possibly regulatory) transcripts
  - new stress-induced genes...

G. Zeller et al., Plant J. (2009)

New Stress-Induced TARs

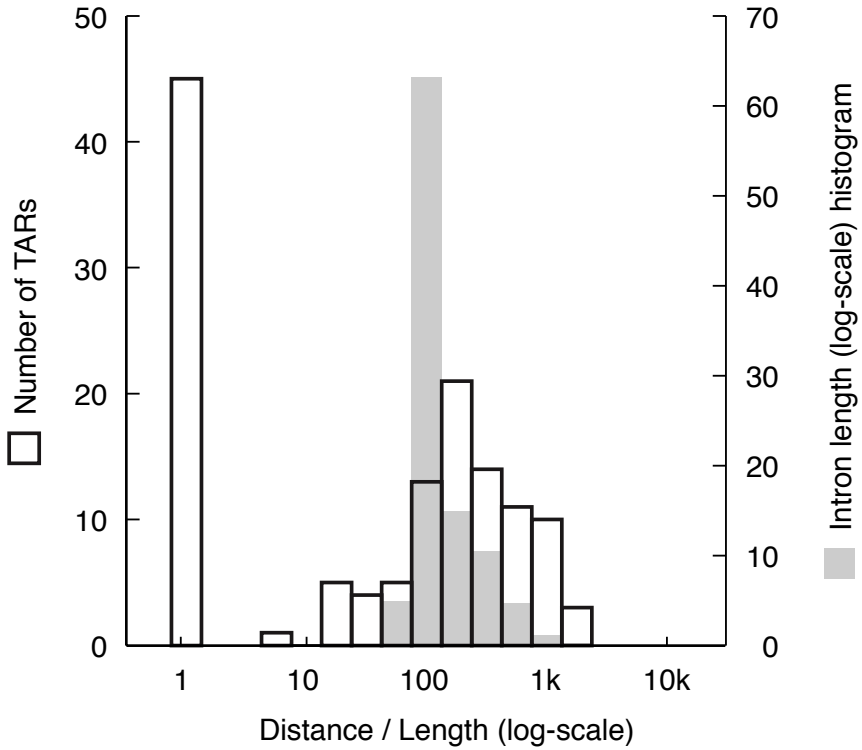


Identify TARs induced under stress (by significant increase of hybridization signal)

- We found candidates for...
  - stress-specific alternative transcripts
  - overlapping (possibly regulatory) transcripts
  - new stress-induced genes...

G. Zeller et al., Plant J. (2009)

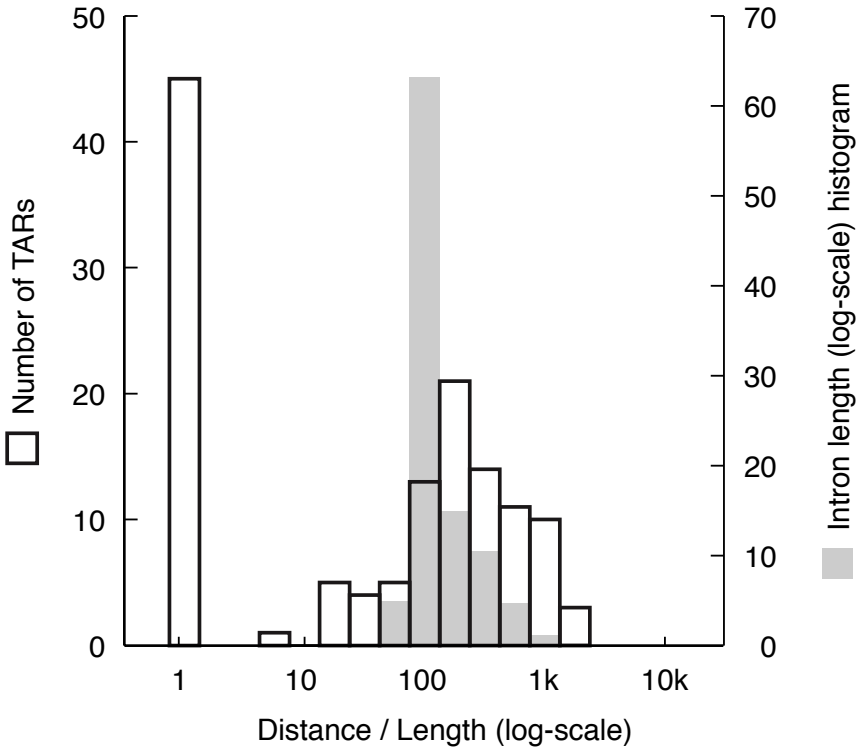
New Stress-Induced Genes

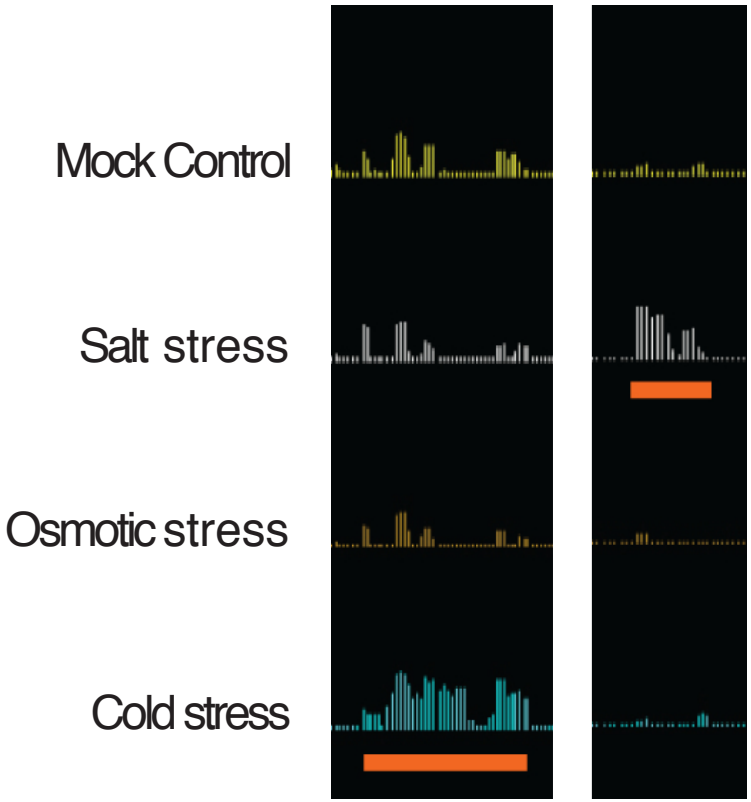


Stress-induced TARs with distance > 500 nt to the nearest annotated gene.

G. Zeller et al., Plant J. (2009)

New Stress-Induced Genes





Stress-induced TARs with distance > 500 nt to the nearest annotated gene.

G. Zeller et al., Plant J. (2009)

Analysis of the *dcl1* Transcriptome



- The DCL1 protein participates in processing miRNA precursor genes into functional miRNAs.
- miRNAs in turn negatively regulate many other genes and are crucial for development and stress response.
- In *dcl1* knock-out mutants, substrate transcripts of DCL1 accumulate and can be detected with tiling arrays.

[Laubinger et al., 2010]

Analysis of the *dcl1* Transcriptome



- The DCL1 protein participates in processing miRNA precursor genes into functional miRNAs.
- miRNAs in turn negatively regulate many other genes and are crucial for development and stress response.
- In *dcl1* knock-out mutants, substrate transcripts of DCL1 accumulate and can be detected with tiling arrays.

[Laubinger et al., 2010]

Analysis of the *dcl1* Transcriptome



- The DCL1 protein participates in processing miRNA precursor genes into functional miRNAs.
- miRNAs in turn negatively regulate many other genes and are crucial for development and stress response.
- In *dcl1* knock-out mutants, substrate transcripts of DCL1 accumulate and can be detected with tiling arrays.

[Laubinger et al., 2010]

Summary



- Normalization improves accuracy of downstream analyses.
- Segmentation of tiling array data with mSTAD...
  - accurately delineates known genes but
  - nevertheless reveals thousands of new transcripts.
- New TARs...
  - include hundreds of stress-specific transcripts
  - detected in mutants of sRNA processing factors include long miRNA precursor genes and transposon transcripts
- Current and future projects include:
  - comparing hybridization-based and sequence-based gene predictions,
  - analyzing *C. elegans* tiling array data (modENCODE project).

Summary



- Normalization improves accuracy of downstream analyses.
- Segmentation of tiling array data with mSTAD...
  - accurately delineates known genes but
  - nevertheless reveals thousands of new transcripts.
- New TARs...
  - include hundreds of stress-specific transcripts
  - detected in mutants of sRNA processing factors include long miRNA precursor genes and transposon transcripts
- Current and future projects include:
  - comparing hybridization-based and sequence-based gene predictions,
  - analyzing *C. elegans* tiling array data (modENCODE project).

Summary



- Normalization improves accuracy of downstream analyses.
- Segmentation of tiling array data with mSTAD...
  - accurately delineates known genes but
  - nevertheless reveals thousands of new transcripts.
- New TARs...
  - include hundreds of stress-specific transcripts
  - detected in mutants of sRNA processing factors include long miRNA precursor genes and transposon transcripts
- Current and future projects include:
  - comparing hybridization-based and sequence-based gene predictions,
  - analyzing *C. elegans* tiling array data (modENCODE project).

Summary



- Normalization improves accuracy of downstream analyses.
- Segmentation of tiling array data with mSTAD...
  - accurately delineates known genes but
  - nevertheless reveals thousands of new transcripts.
- New TARs...
  - include hundreds of stress-specific transcripts
  - detected in mutants of sRNA processing factors include long miRNA precursor genes and transposon transcripts
- Current and future projects include:
  - comparing hybridization-based and sequence-based gene predictions,
  - analyzing *C. elegans* tiling array data (modENCODE project).

Acknowledgments

We gratefully acknowledge help with the material:

- Ali Mortazavi, Caltech
- Jonas Behr, Friedrich Miescher Laboratory
- Gabriele Schweikert, Friedrich Miescher Laboratory
- Sebastian Schultheiss, Friedrich Miescher Laboratory
- Christian Widmer, Friedrich Miescher Laboratory
- Sören Sonnenburg, Berlin Institute of Technology
- Georg Zeller, EMBL Heidelberg

The slides with references will be available online at

`http://www.fml.mpg.de/raetsch/lectures/erice10`

References I

J. Behr, G. Schweikert, J. Cao, F. De Bona, G. Zeller, S. Laubinger, S. Ossowski, K. Schneeberger, D. Weigel, and G. Rätsch. Rna-seq and tiling arrays for improved gene finding. Oral presentation at the CSHL Genome Informatics Meeting, September 2008. URL `http://www.fml.tuebingen.mpg.de/raetsch/lectures/RaetschGenomeInformatics08.pdf`.

J. Behr, G. Schweikert, J. Cao, F. De Bona, G. Zeller, S. Laubinger, S. Ossowski, K. Schneeberger, D. Weigel, and G. Rätsch. Rna-seq and tiling arrays for improved gene finding. Presented at the CSHL Genome Informatics Meeting, July 2009, in prep.

M. Bergkessel, G. Wilmes, and C. Guthrie. Snapshot: Formation of mrnps. *Cell*, 136, January 2009.

B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.

RM Clark, G Schweikert, C Toomajian, S Ossowski, G Zeller, P Shinn, N Warthmann, TT Hu, G Fu, DA Hinds, H Chen, KA Frazer, DH Huson, B Schölkopf, M Nordborg, G Rätsch, JR Ecker, and D Weigel. Common sequence polymorphisms shaping genetic diversity in arabidopsis thaliana. *Science*, 317(5836):338–342, 2007. ISSN 1095-9203 (Electronic). doi: 10.1126/science.1138632.

References II

A. Coghlan, T.J. Fiedler, S.J. McKay, P. Flicek, T.W. Harris, D. Blasiar, The nGASP Consortium, and L.D. Stein. ngasp: the nematode genome annotation assessment project. *BMC Bioinformatics*, 2008. submitted.

C. Cortes and V.N. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

H. Daumé. Frustratingly easy domain adaptation. In *ACL*. The Association for Computer Linguistics, 2007. URL `http://aclweb.org/anthology-new/P/P07/P07-1033.pdf`.

Juliane C Dohm, Claudio Lottaz, Tatiana Borodina, and Heinz Himmelbauer. Sharcgs, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. *Genome Res*, 17(11):1697–706, Nov 2007. doi: 10.1101/gr.6435207.

T. Evgeniou, C.A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

S.S. Gross, O. Russakovsky, C.B. Do, and S. Batzoglou. Training conditional random fields for maximum labelwise accuracy. In *Advances in Neural Information Processing Systems 19 (NIPS 2006)*, 2007.

T.S. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *J. Comp. Biol.*, 7:95–114, 2000.

L. Jacob and J.P. Vert. Efficient peptide-MHC-I binding prediction for alleles with few known binders. *Bioinformatics*, 24(3):358, 2008.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proc. 18th Int. Conf. Mach. Learn.*, 2001.

References III

L. Liao and W.S. Noble. Combining pairwise sequence similarity and support vector machines. In *Proc. 6th Int. Conf. Computational Molecular Biology*, pages 225–232, 2002.

Iain Maccallum, Dariusz Przybylski, Sante Gnerre, Joshua Burton, Ilya Shlyakhter, Andreas Gnirke, Joel Malek, Kevin McKernan, Swati Ranade, Terrance P Shea, Louise Williams, Sarah Young, Chad Nusbaum, and David B Jaffe. Allpaths 2: small genomes assembled accurately and with high continuity from short paired reads. *Genome Biol*, 10(10):R103, 2009. doi: 10.1186/gb-2009-10-10-r103.

K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

P A Pevzner, H Tang, and M S Waterman. An eulerian path approach to dna fragment assembly. *Proc Natl Acad Sci U S A*, 98(17):9748–53, Aug 2001. doi: 10.1073/pnas.171285098.

G. Rätsch and S. Sonnenburg. Accurate splice site detection for *Caenorhabditis elegans*. In K. Tsuda B. Schoelkopf and J.-P. Vert, editors, *Kernel Methods in Computational Biology*. MIT Press, 2004.

G. Rätsch, S. Sonnenburg, and B. Schölkopf. RASE: recognition of alternatively spliced exons in *C. elegans*. *Bioinformatics*, 21(Suppl. 1):i369–i377, June 2005.

B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

References IV

S.J. Schultheiss, W. Busch, J.U. Lohmann, O. Kohlbacher, and G. Rätsch. Kirmes: Kernel-based identification of regulatory modules in euchromatic sequences. In *German Conference on Bioinformatics*, Lecture notes in Informatics, pages 158–167, Heidelberg, 2008. GI, Springer Verlag. URL `http://www.fml.tuebingen.mpg.de/raetsch/projects/kirmes`.

Gabriele Schweikert, Jonas Behr, Alexander Zien, Georg Zeller, Cheng Soon Ong, Sören Sonnenburg, and Gunnar Rätsch. mgene.web: a web service for accurate computational gene finding. *Nucleic Acids Research*, Web Server Issue, 2009a. URL `http://mgene.org/web`. Advance Access published on June 3, 2009.

Gabriele Schweikert, Alexander Zien, Georg Zeller, Jonas Behr, Christoph Dieterich, Cheng Soon Ong, Petra Philips, Fabio De Bona, Lisa Hartmann, Anja Bohlen, Nina Krüger, Sören Sonnenburg, and Gunnar Rätsch. mgene: Accurate svm-based gene finding with an application to nematode genomes. *Genome Research*, 2009b. URL `http://genome.cshlp.org/content/early/2009/06/29/gr.090597.108.full.pdf+html`. Advance access June 29, 2009.

S. Sonnenburg, G. Rätsch, A. Jagota, and K.-R. Müller. New methods for splice-site recognition. In *Proc. International Conference on Artificial Neural Networks*, 2002.

S. Sonnenburg, G. Rätsch, and K. Rieck. Large scale learning with string kernels. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines*. MIT Press, 2007a.

References V

S Sonnenburg, G Schweikert, P Philips, J Behr, and G Rätsch. Accurate splice site prediction using support vector machines. *BMC Bioinformatics*, 8 Suppl 10:S7, 2007b. ISSN 1471-2105 (Electronic). doi: 10.1186/1471-2105-8-S10-S7.

Sören Sonnenburg, Alexander Zien, and Gunnar Rätsch. ARTS: Accurate Recognition of Transcription Starts in Human. *Bioinformatics*, 22(14):e472–480, 2006.

K. Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K.R. Müller. A new discriminative kernel from probabilistic models. *Neural Computation*, 14:2397–2414, 2002.

V.N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, New York, 1995.

J.-P. Vert, H. Saigo, and T. Akutsu. Local alignment kernels for biological sequences. In K. Tsuda B. Schoelkopf and J.-P. Vert, editors, *Kernel Methods in Computational Biology*. MIT Press, 2004.

René L Warren, Granger G Sutton, Steven J M Jones, and Robert A Holt. Assembling millions of short dna sequences using ssake. *Bioinformatics*, 23(4):500–1, Feb 2007. doi: 10.1093/bioinformatics/btl629.

C. Widmer, J. Leiva, Y. Altun, and G. Rätsch. Leveraging sequence classification by taxonomy-based multitask learning. In *Proc. RECOMB'10.*, 2010.

G Zeller, RM Clark, K Schneeberger, A Bohlen, D Weigel, and G Ratsch. Detecting polymorphic regions in arabidopsis thaliana with resequencing microarrays. *Genome Res*, 18 (6):918–929, 2008. ISSN 1088-9051 (Print). doi: 10.1101/gr.070169.107.

References VI

D.R. Zerbino and E. Birney. Velvet: Algorithms for de novo short read assembly using de bruijn graphs. *Genome Research*, 18:828–829, 2008.

A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K.-R. Müller. Engineering Support Vector Machine Kernels That Recognize Translation Initiation Sites. *BioInformatics*, 16(9): 799–807, September 2000.