Large Scale Sequence Analysis with Applications to Genomics

Gunnar Rätsch

Friedrich Miescher Laboratory, Max Planck Society
Tübingen, Germany

Talk at CWI, Amsterdam

October 23, 2009

Discovery of the Nuclein (Friedrich Miescher, 1869)









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)

Research Topics

Machine Learning

1. Statistical inference methods for structured data
   ⇒ *Develop fast and accurate learning methods*
2. Convergence properties of iterative algorithms
   ⇒ *Boosting-like algorithms and semi-infinite LPs*

Molecular Biology

3. Genome annotation
   ⇒ *Predict features encoded on DNA*
4. Biological networks
   ⇒ *Understand interactions between gene products*
5. Analysis of polymorphisms
   ⇒ *Discover polymorphisms and associate with phenotypes*

Research Topics

Machine Learning

1. Statistical inference methods for structured data
   ⇒ *Develop fast and accurate learning methods*
2. Convergence properties of iterative algorithms
   ⇒ *Boosting-like algorithms and semi-infinite LPs*

Molecular Biology

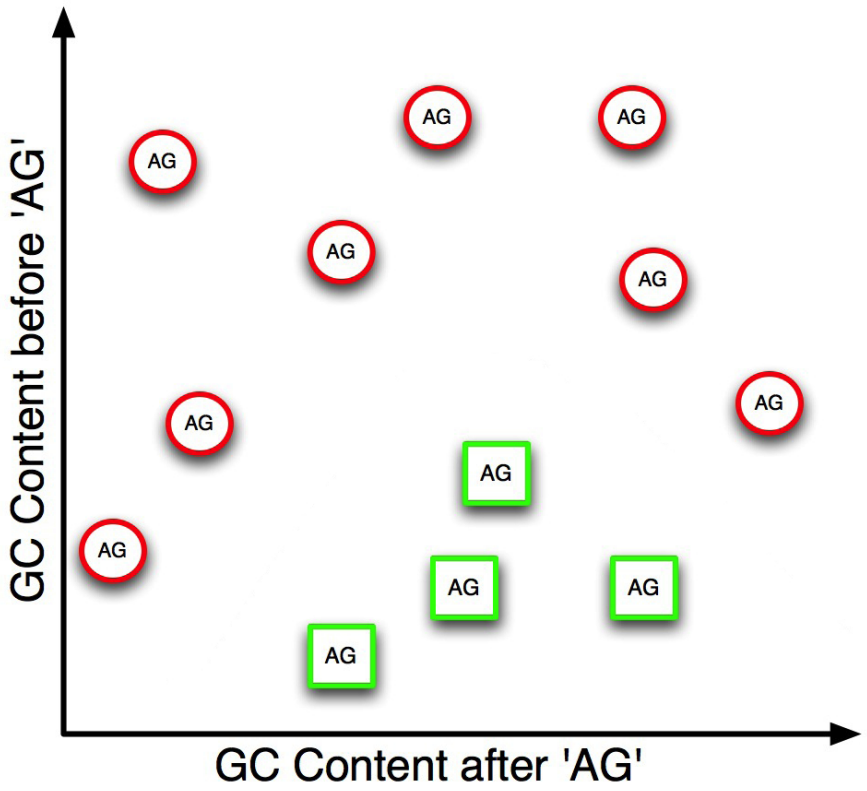
3. Genome annotation
   ⇒ *Predict features encoded on DNA*
4. Biological networks
   ⇒ *Understand interactions between gene products*
5. Analysis of polymorphisms
   ⇒ *Discover polymorphisms and associate with phenotypes*

Machine Learning

Given: Observations of some complex phenomenon
Goal: Learn from data & build predictive models

Given: Observations of some complex phenomenon
Goal: Learn from data & build predictive models

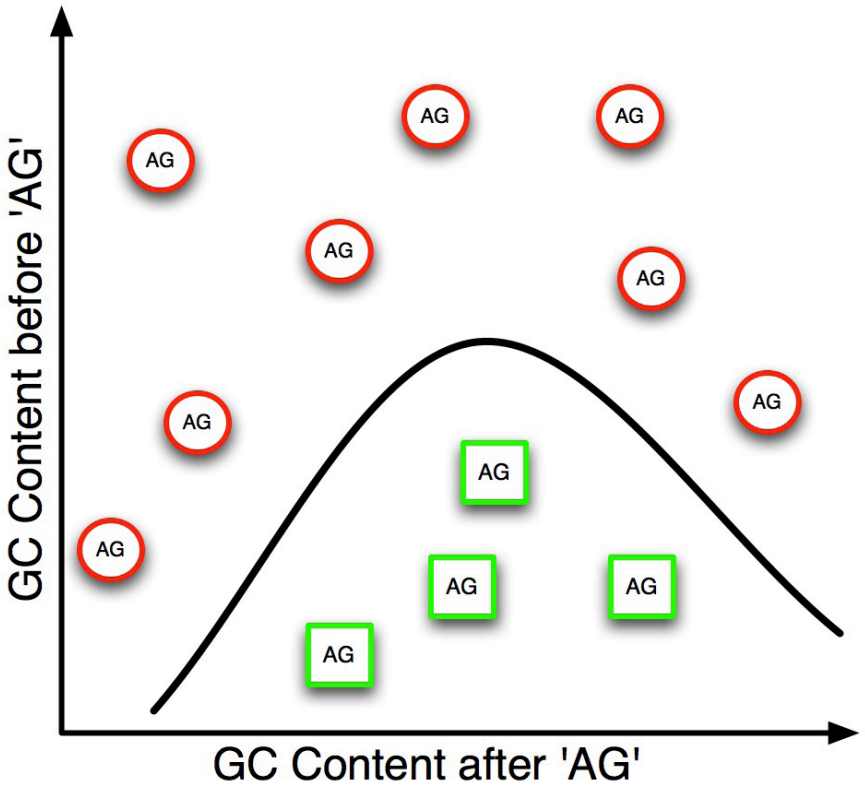
Example:



Two different classes of observations

Given: Observations of some complex phenomenon
Goal: Learn from data & build predictive models

Example:

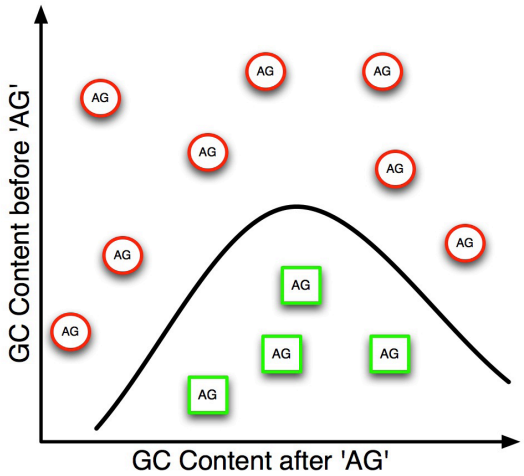


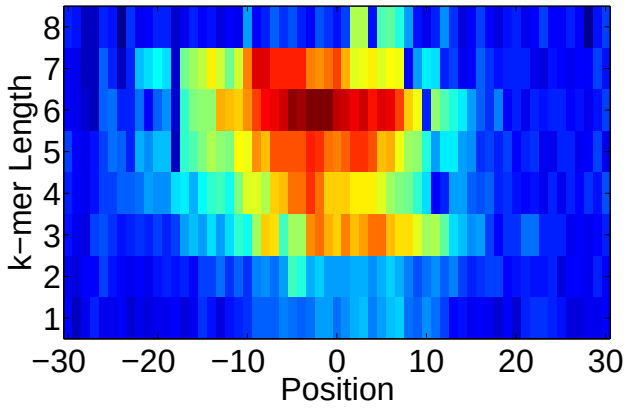
Inferred classification rule

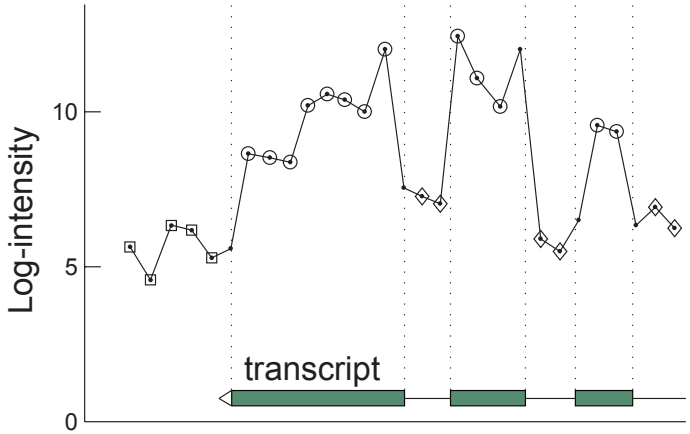
Given: Observations of some complex phenomenon
Goal: Learn from data & build predictive models

Recent contributions:

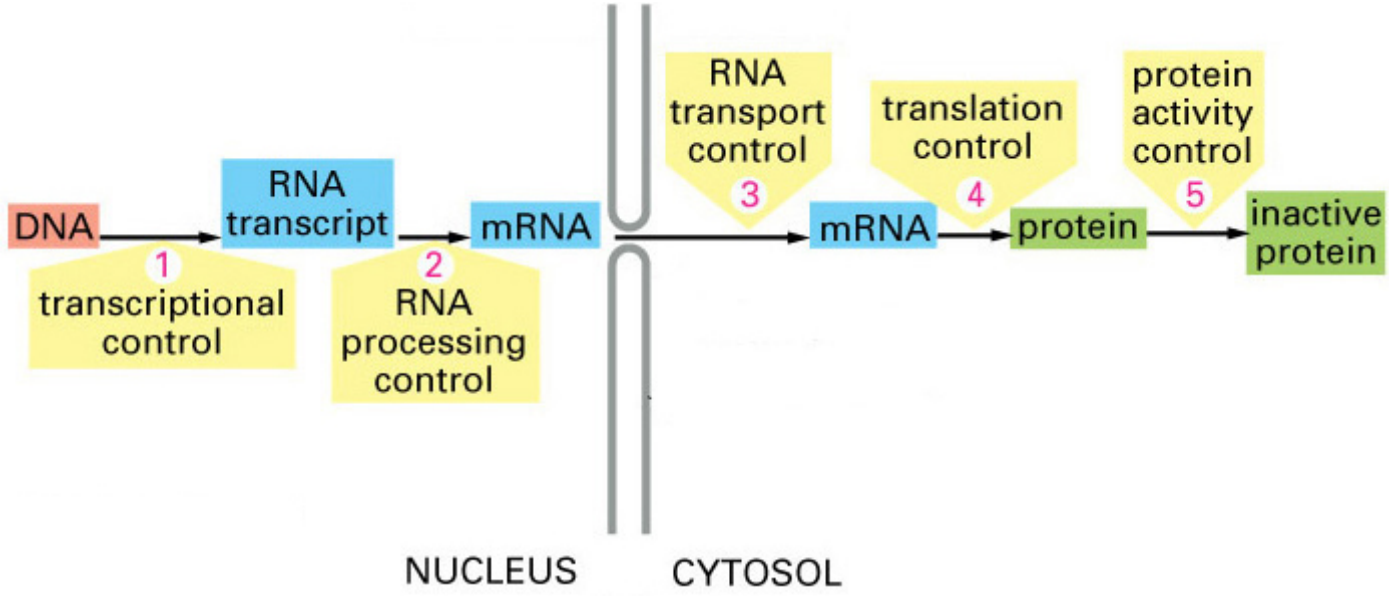
1. Large scale sequence classification
2. Analysis and explanation of learning results
3. Sequence segmentation & structure prediction







Learning about the Transcriptome



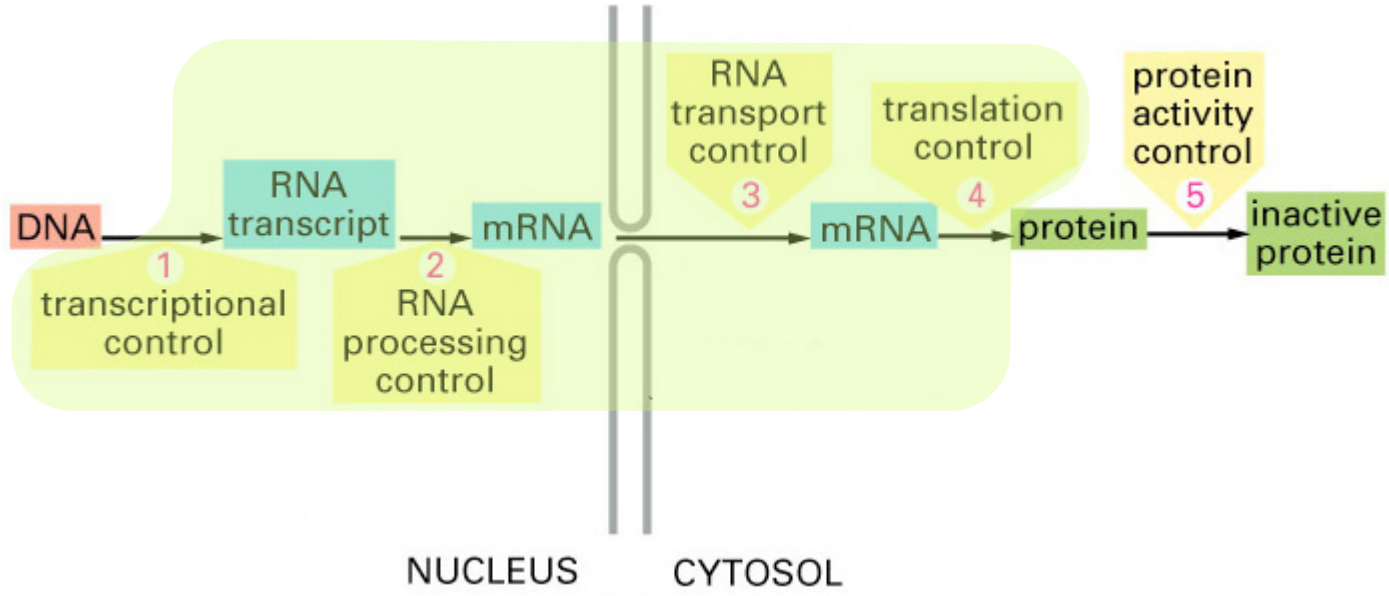
Machine Learning View

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

Biological View

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

Learning about the Transcriptome



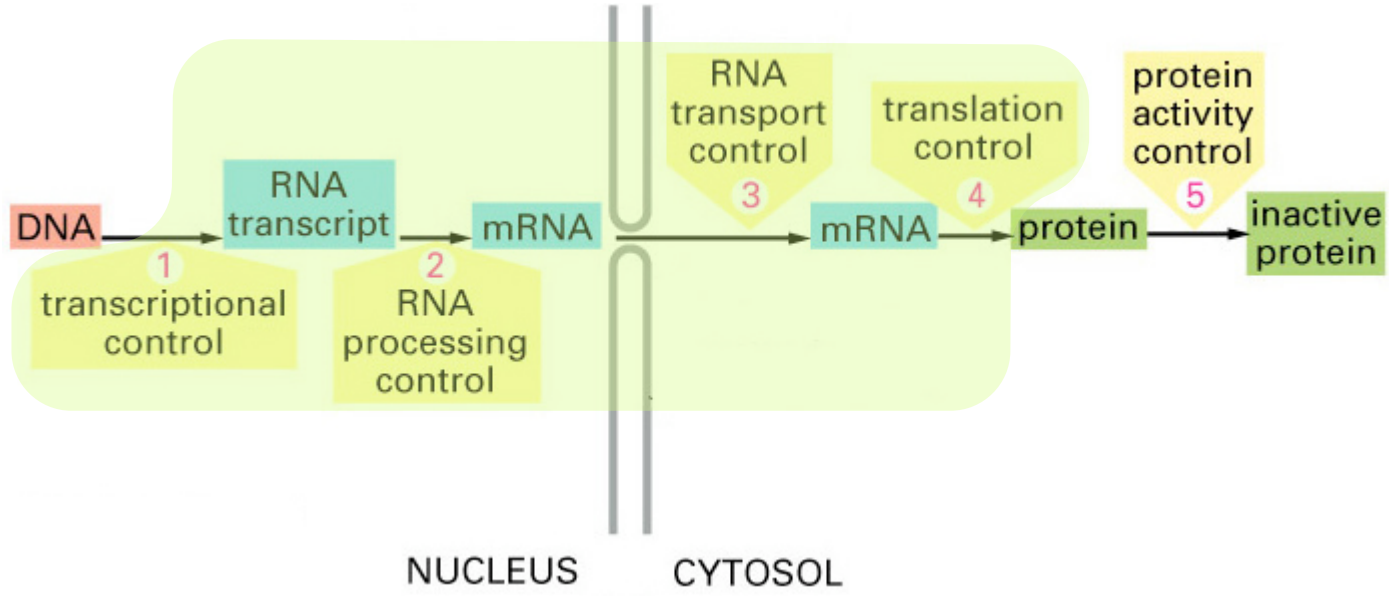
Machine Learning View

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

Biological View

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

Learning about the Transcriptome



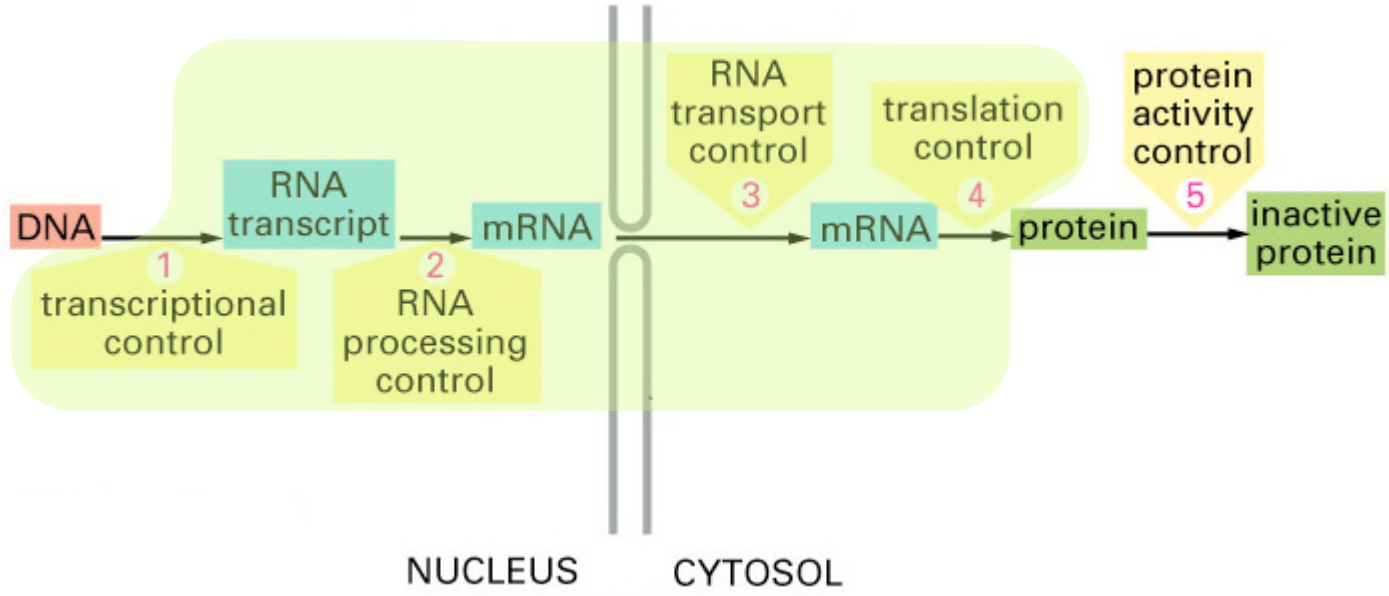
Machine Learning View

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

Biological View

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

Learning about the Transcriptome



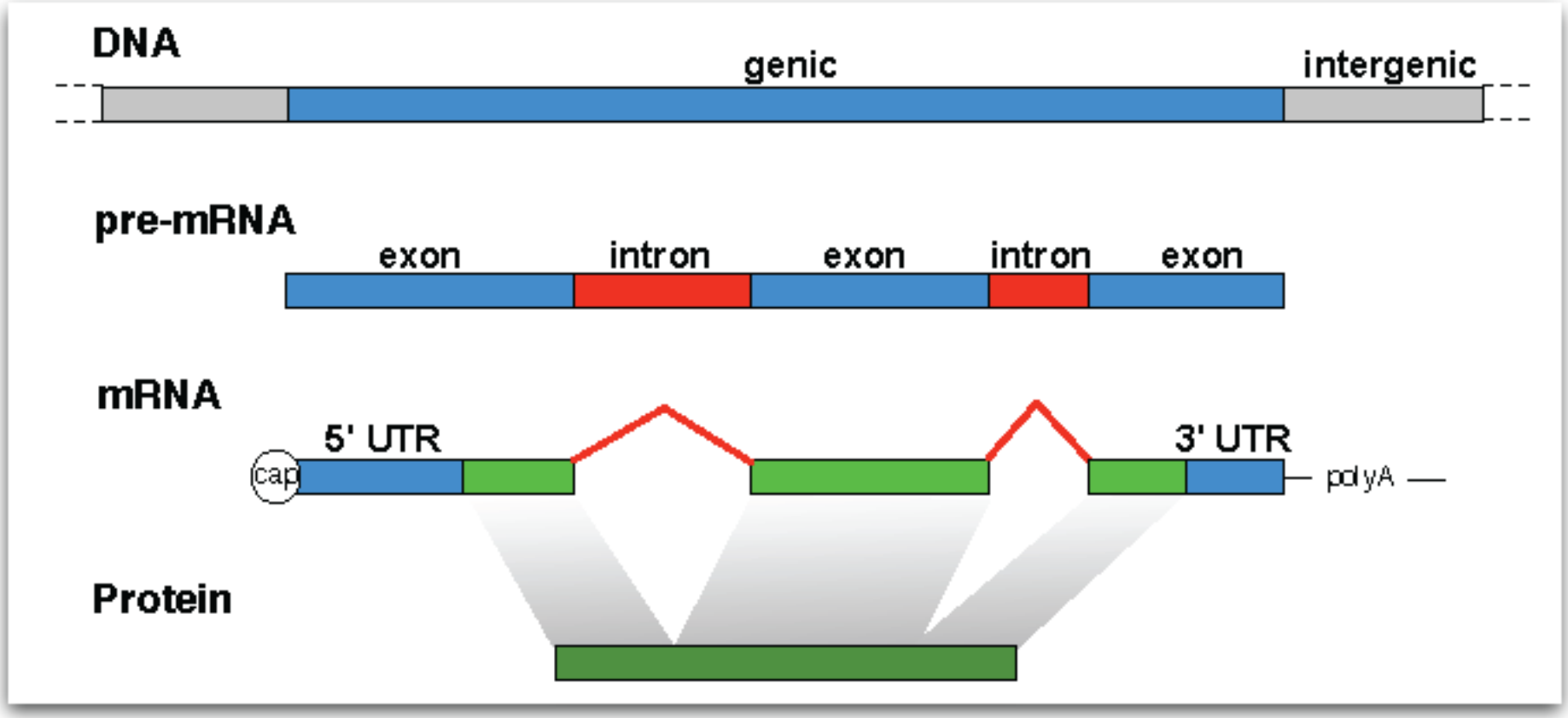
Machine Learning View

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

Biological View

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

Computational Genome Annotation



Simplest formulation:

- Given a DNA sequence $\mathbf{x} \in \{\text{'A','C','G','T'}\}^L$
- Find the correct **label sequence y** $= y_1 y_2 \dots y_L$
  ($y_i \in Y = \{$"intergenic", "5' UTR", "coding", "intron", $\dots\}$)

Example: *C. elegans* (I: 43,500 - 52,050)

Friedrich Miescher Laboratory
of the Max Planck Society



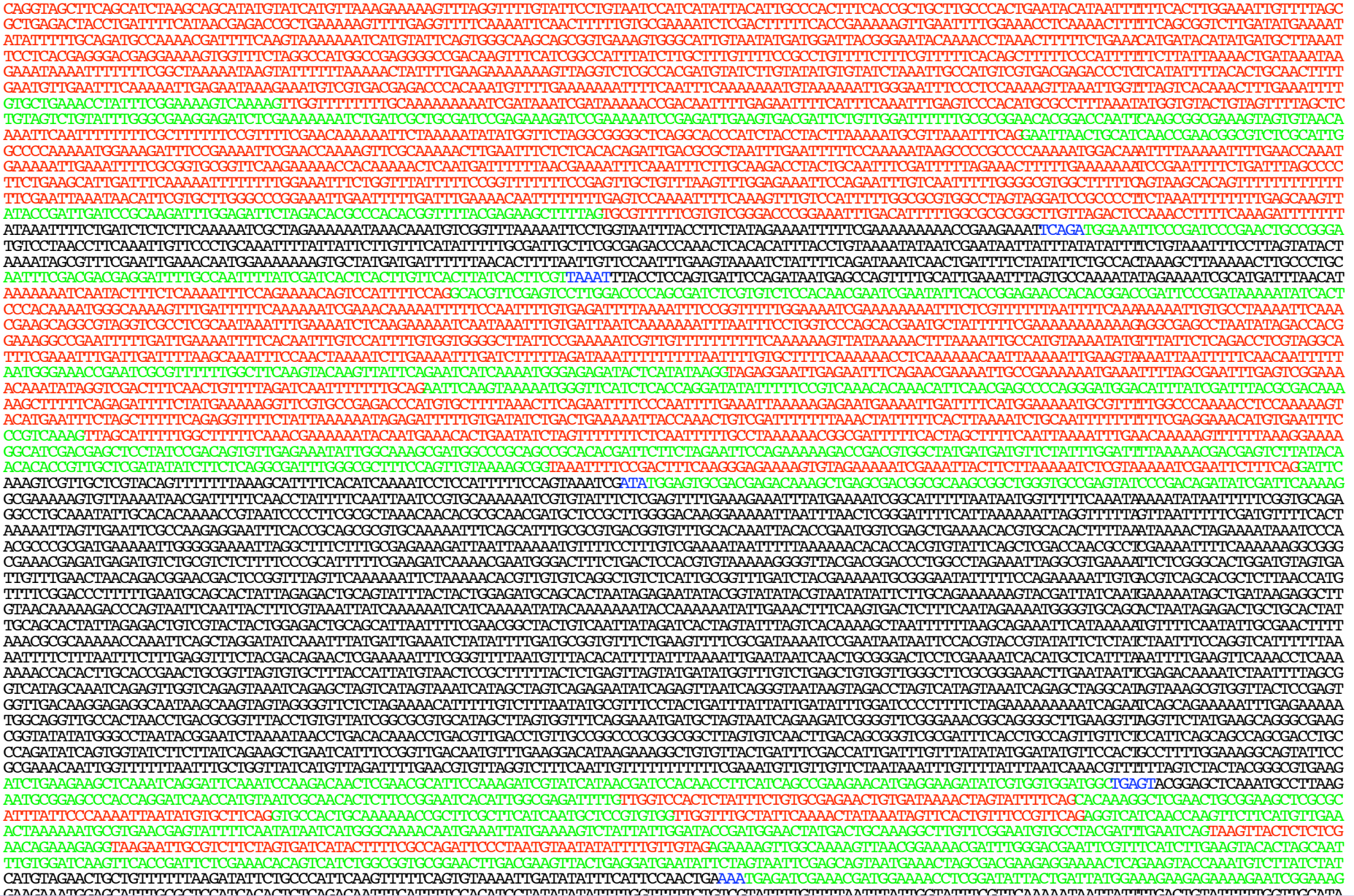
Example: *C. elegans* (I: 43,500 - 52,050)

Friedrich Miescher Laboratory
of the Max Planck Society



Example: *C. elegans* (I: 43,500 - 52,050)

Friedrich Miescher Laboratory
of the Max Planck Society



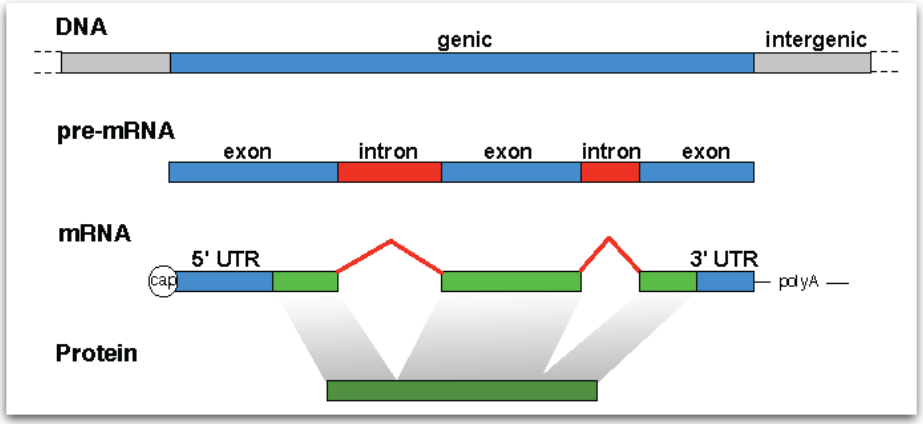
Quasi-Standard: Hidden Markov Models

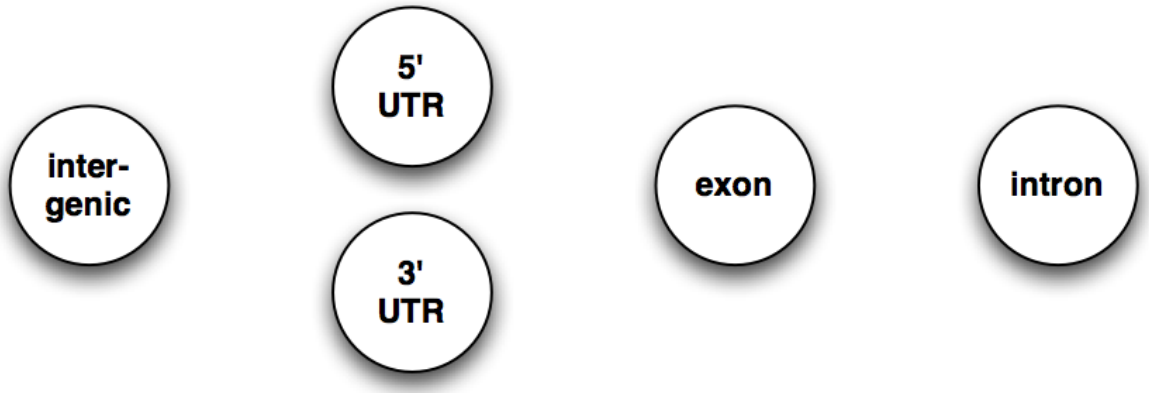
Model sequence content:

- One state per segment type
- Allow only plausible transitions
- Content statistics at each state
  - Derived from known genes

Prediction:

- Given DNA, find most likely state sequences

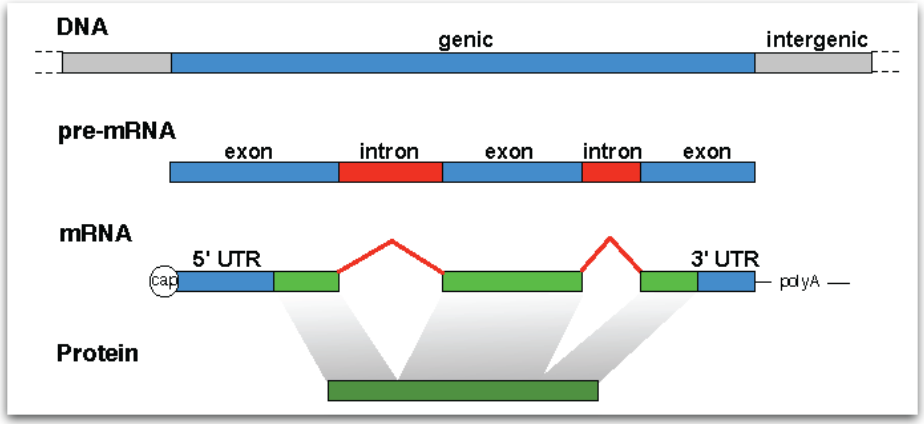




Quasi-Standard: Hidden Markov Models

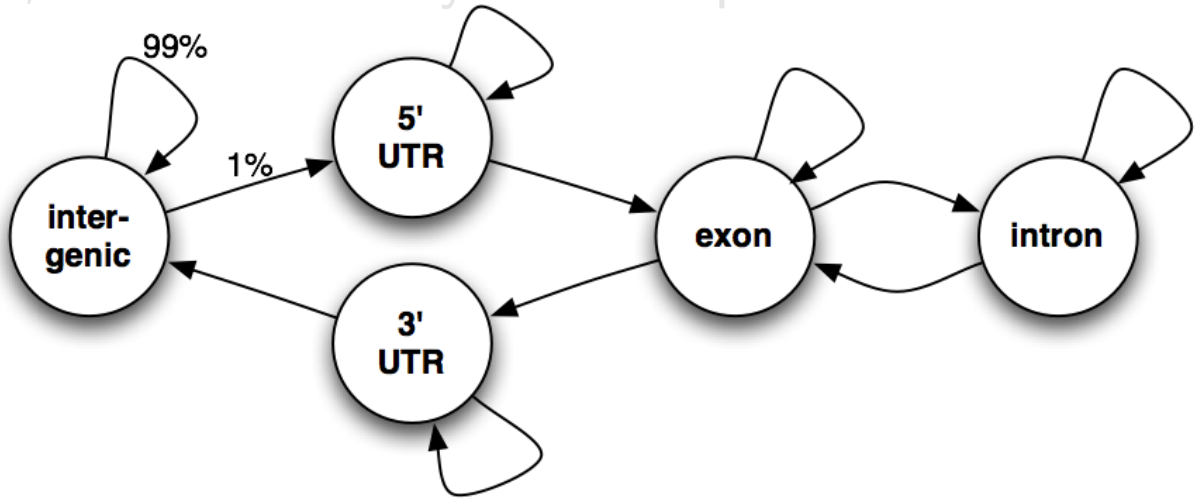
Model sequence content:

- One state per segment type
- Allow only plausible transitions
- Content statistics at each state
  - Derived from known genes



Prediction:

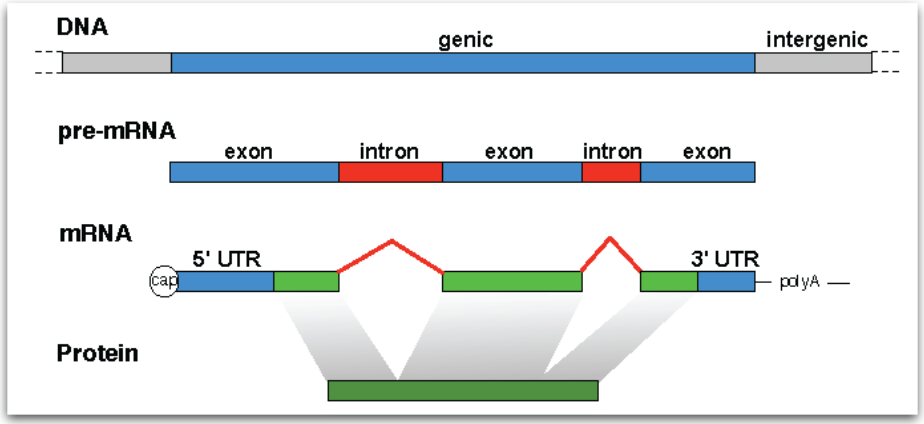
- Given DNA, find most likely state sequences



Quasi-Standard: Hidden Markov Models

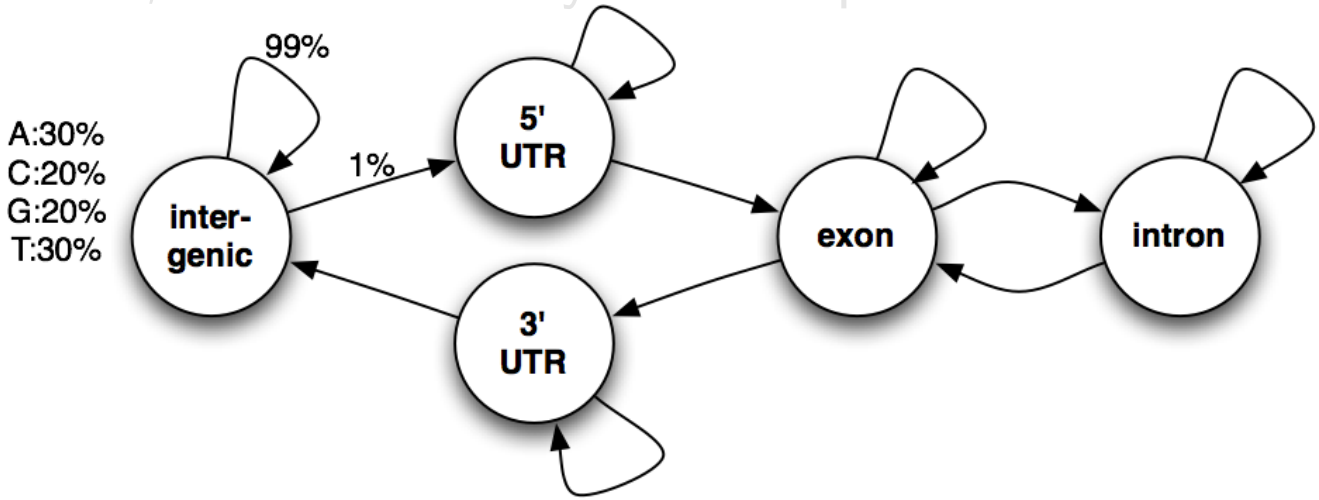
Model sequence content:

- One state per segment type
- Allow only plausible transitions
- Content statistics at each state
  - Derived from known genes



Prediction:

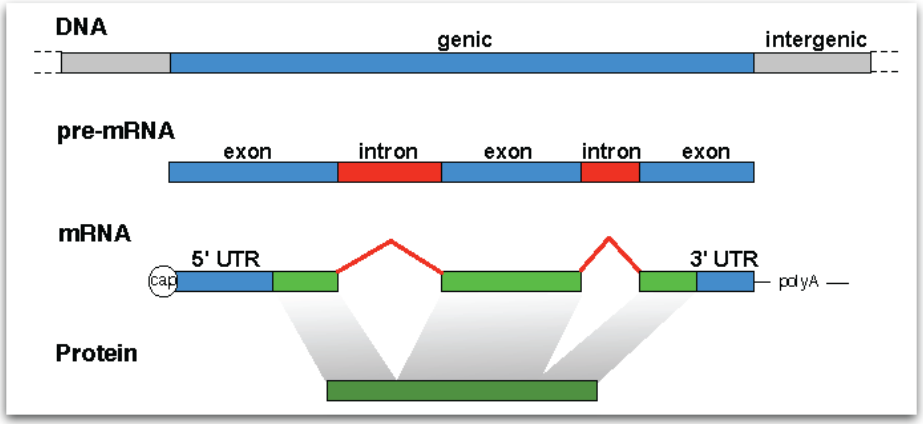
- Given DNA, find most likely state sequences



Quasi-Standard: Hidden Markov Models

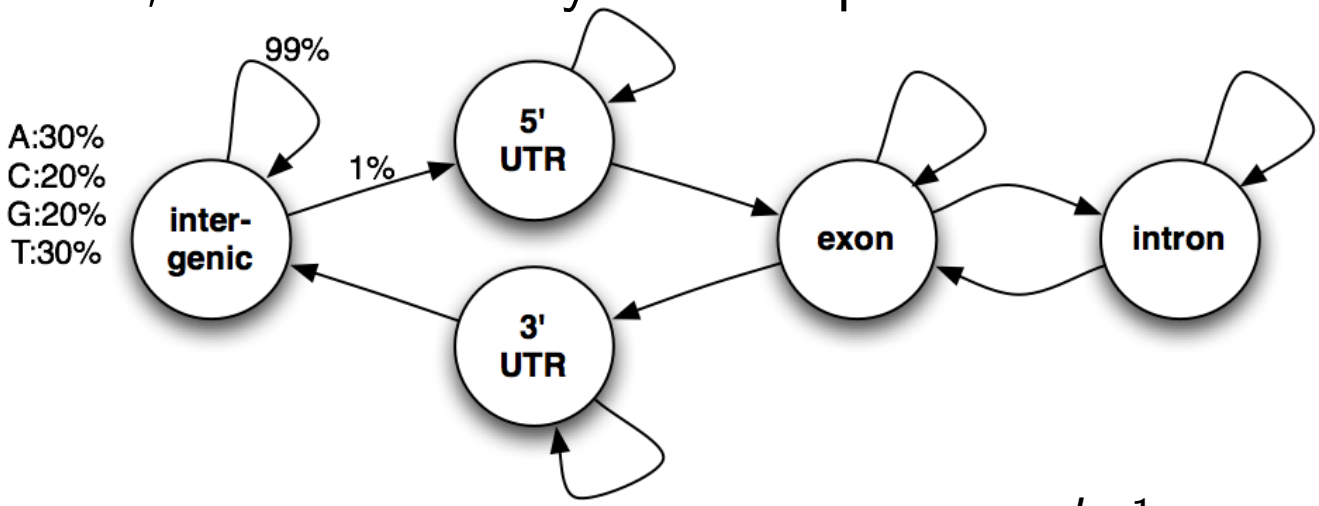
Model sequence content:

- One state per segment type
- Allow only plausible transitions
- Content statistics at each state
  - Derived from known genes



Prediction:

- Given DNA, find most likely state sequences

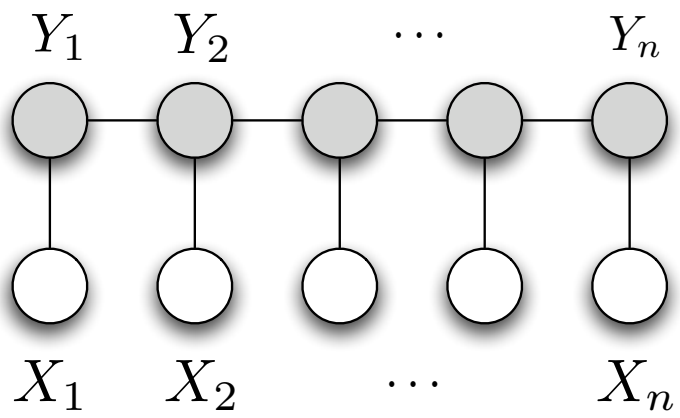


$$p(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^{L-1} p(x_i|y_i)p(y_{i+1}|y_i)$$

Hidden Markov Models [Rabiner, 1989]

- State sequence treated as (first order) Markov chain
- No direct dependencies between observations

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



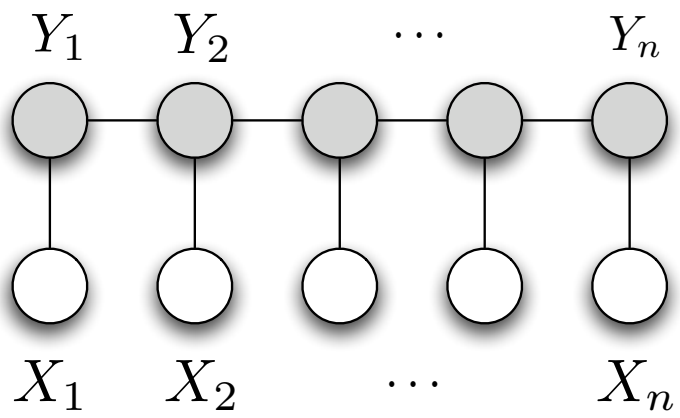
- Maximum likelihood estimation: $\max_\theta \sum_{n=1}^{N} \log p(\mathbf{x}_n, \mathbf{y}_n)$
- Decoding: Viterbi/dynamic programming (DP) algorithms

Generative Models

Hidden Markov Models [Rabiner, 1989]

- State sequence treated as (first order) Markov chain
- No direct dependencies between observations

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



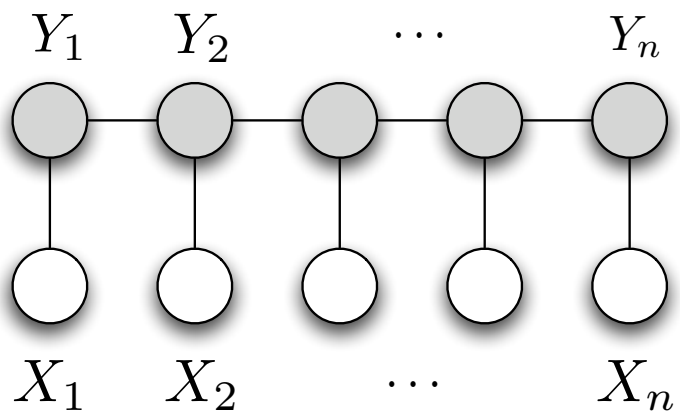
- Maximum likelihood estimation: $\max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log p(\mathbf{x}_n, \mathbf{y}_n)$
- Decoding: Viterbi/dynamic programming (DP) algorithms

Generative Models

Hidden Markov Models [Rabiner, 1989]

- State sequence treated as (first order) Markov chain
- No direct dependencies between observations

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



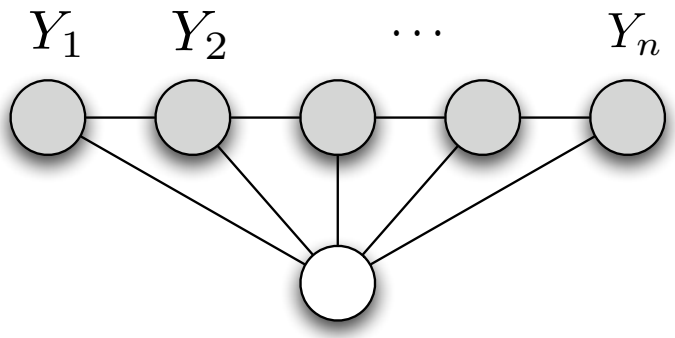
- Maximum likelihood estimation: $\max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log p(\mathbf{x}_n, \mathbf{y}_n)$
- Decoding: Viterbi/dynamic programming (DP) algorithms

Discriminative Models

Conditional Random Fields [Lafferty et al., 2001]

- conditional probability $p(y|x)$ instead of joint probability $p(x, y)$

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



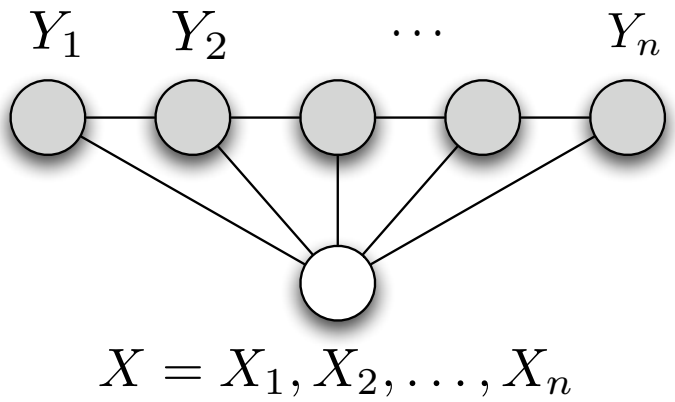
- Can handle non-independent input features
- Maximum likelihood estimation: $\max_{\mathbf{w}} \sum_{n=1}^{N} \log p_{\mathbf{w}}(\mathbf{y}_n|\mathbf{x}_n)$
- Decoding: Viterbi or MEA algorithms [Gross et al., 2006]

Discriminative Models

Conditional Random Fields [Lafferty et al., 2001]

- conditional probability $p(y|x)$ instead of joint probability $p(x, y)$

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



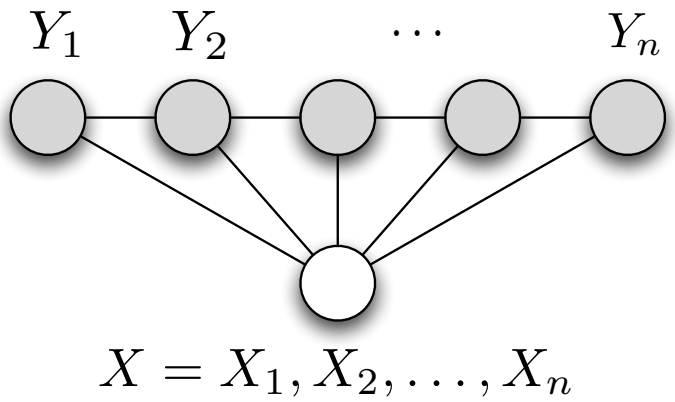
- Can handle non-independent input features
- Maximum likelihood estimation: $\max_{\mathbf{w}} \sum_{n=1}^{N} \log p_{\mathbf{w}}(\mathbf{y}_n|\mathbf{x}_n)$
- Decoding: Viterbi or MEA algorithms [Gross et al., 2006]

Discriminative Models

Conditional Random Fields [Lafferty et al., 2001]

- conditional probability $p(y|x)$ instead of joint probability $p(x, y)$

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



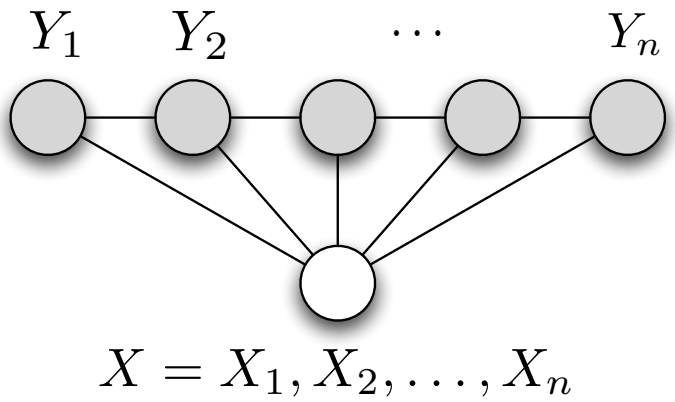
- Can handle non-independent input features
- Maximum likelihood estimation: $\max_{\mathbf{w}} \sum_{n=1}^{N} \log p_{\mathbf{w}}(\mathbf{y}_n|\mathbf{x}_n)$
- Decoding: Viterbi or MEA algorithms [Gross et al., 2006]

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
- Maximum likelihood estimation: $\max_{\mathbf{w}} \sum_{n=1}^{N} \log p_{\mathbf{w}}(\mathbf{y}_n|\mathbf{x}_n)$
- Decoding: Viterbi or MEA algorithms [Gross et al., 2006]

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:

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

- Idea: $f(\mathbf{y}|\mathbf{x}) \gg f(\hat{\mathbf{y}}|\mathbf{x})$ for wrong labels $\hat{\mathbf{y}} \neq \mathbf{y}$
- Max-margin parameter estimation: [Altun et al., 2003]
  - Given N sequence pairs $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$ for training
  - Solve optimization problem:

$$\max_{f,\rho\in\mathbb{R},\xi\in\mathbb{R}^N_+} \quad \rho - C\sum_{n=1}^{N}\xi_n$$

$$\text{w.r.t.} \quad f(\mathbf{y}_n|\mathbf{x}_n) - f(\mathbf{y}|\mathbf{x}_n) \geq \rho - \xi_n$$

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

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:

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

- Idea: $f(\mathbf{y}|\mathbf{x}) \gg f(\hat{\mathbf{y}}|\mathbf{x})$ for wrong labels $\hat{\mathbf{y}} \neq \mathbf{y}$
- Max-margin parameter estimation: [Altun et al., 2003]
  - Given N sequence pairs $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$ for training
  - Solve optimization problem:

$$\begin{aligned} \max_{f,\rho\in\mathbb{R},\boldsymbol{\xi}\in\mathbb{R}^N_+} \quad & \rho - C\sum_{n=1}^{N}\xi_n \\ \text{w.r.t.} \quad & f(\mathbf{y}_n|\mathbf{x}_n) - f(\mathbf{y}|\mathbf{x}_n) \geq \rho - \xi_n \\ & \text{for all } \mathbf{y}_n \neq \mathbf{y} \in \Upsilon^*, n = 1, \ldots, N \end{aligned}$$

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:

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

- Idea: $f(\mathbf{y}|\mathbf{x}) \gg f(\hat{\mathbf{y}}|\mathbf{x})$ for wrong labels $\hat{\mathbf{y}} \neq \mathbf{y}$
- Max-margin parameter estimation: [Altun et al., 2003]
  - Given N sequence pairs $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$ for training
  - Solve optimization problem:

$$\begin{aligned} \max_{f,\rho\in\mathbb{R},\boldsymbol{\xi}\in\mathbb{R}^N_+} \quad & \rho - C\sum_{n=1}^{N}\xi_n \\ \text{w.r.t.} \quad & f(\mathbf{y}_n|\mathbf{x}_n) - f(\mathbf{y}|\mathbf{x}_n) \geq \rho - \xi_n \\ & \text{for all } \mathbf{y}_n \neq \mathbf{y} \in \Upsilon^*, n = 1, \ldots, N \end{aligned}$$

Structured Output Learning with $\Phi(\mathbf{x}, \mathbf{y})$

- Assume $f(\mathbf{y}|\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$, where $\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \in \mathcal{F}$, $\|\mathbf{w}\|_2 = 1$

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

- Linear classifier that separates true from wrong labellings
- **Representer Theorem:** $\mathbf{w} = \sum_{n=1}^{N} \sum_{\mathbf{y} \in \Upsilon^*} \alpha_{n,\mathbf{y}} \Phi(\mathbf{x}_n, \mathbf{y})$
- Corollary: (1) only depend on products of Φ's
- Define "kernel function":

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

$\Rightarrow$ "Kernel Trick" (everywhere in the machine learning)

Structured Output Learning with $\Phi(\mathbf{x}, \mathbf{y})$

- Assume $f(\mathbf{y}|\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$, where $\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \in \mathcal{F}$, $\|\mathbf{w}\|_2 = 1$

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

- Linear classifier that separates true from wrong labellings
- **Representer Theorem:** $\mathbf{w} = \sum_{n=1}^{N} \sum_{\mathbf{y} \in \Upsilon^*} \alpha_{n,\mathbf{y}} \Phi(\mathbf{x}_n, \mathbf{y})$
- Corollary: (1) only depend on products of Φ's
- Define "kernel function":

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

$\Rightarrow$ "Kernel Trick" (only have to be able to compute k)

Structured Output Learning with $\Phi(\mathbf{x}, \mathbf{y})$

- Assume $f(\mathbf{y}|\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$, where $\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \in \mathcal{F}$, $\|\mathbf{w}\|_2 = 1$

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

- Linear classifier that separates true from wrong labellings
- **Representer Theorem:** $\mathbf{w} = \sum_{n=1}^{N} \sum_{\mathbf{y} \in \Upsilon^*} \alpha_{n,\mathbf{y}} \Phi(\mathbf{x}_n, \mathbf{y})$
- Corollary: (1) only depend on products of Φ's
- Define "kernel function":

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

$\Rightarrow$ "Kernel Trick" (everything only depends on k)

Structured Output Learning with $\Phi(\mathbf{x}, \mathbf{y})$

- Assume $f(\mathbf{y}|\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$, where $\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \in \mathcal{F}$, $\|\mathbf{w}\|_2 = 1$

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

- Linear classifier that separates true from wrong labellings
- **Representer Theorem:** $\mathbf{w} = \sum_{n=1}^{N} \sum_{\mathbf{y} \in \Upsilon^*} \alpha_{n,\mathbf{y}} \Phi(\mathbf{x}_n, \mathbf{y})$
- Corollary: (1) only depend on products of Φ's
- Define "kernel function":

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

$\Rightarrow$ "Kernel Trick" (everything only depends on k)

Structured Output Learning with $\Phi(\mathbf{x}, \mathbf{y})$

- Assume $f(\mathbf{y}|\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$, where $\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \in \mathcal{F}$, $\|\mathbf{w}\|_2 = 1$

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

- Linear classifier that separates true from wrong labellings
- **Representer Theorem:** $\mathbf{w} = \sum_{n=1}^{N} \sum_{\mathbf{y} \in \Upsilon^*} \alpha_{n,\mathbf{y}} \Phi(\mathbf{x}_n, \mathbf{y})$
- Corollary: (1) only depend on products of Φ's
- Define "kernel function":

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

$\Rightarrow$ **"Kernel Trick" (everything only depends on k)**

Optimization Strategy

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

- Big: one constraint per example and wrong labeling
- Iterative solution
  - Begin with small set of wrong labellings
  - Solve reduced optimization problem
  - Find labellings that violate constraints
  - Add constraints, resolve
- Guaranteed Convergence:
  - $\|\cdot\|_2$: see Tsochantaridis et al. [2005]
  - $\|\cdot\|_1$: see Rätsch and Warmuth [2005] ($\log(N)/\epsilon^2$ iterations)

How to find violated constraints?

- Constraint

$$\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}_n) - \Phi(\mathbf{x}, \mathbf{y}) \rangle \geq \rho - \xi_n$$

- Find labeling $\mathbf{y}$ that maximizes

$$\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$$

- Use Dynamic Programming Decoding

$$\mathbf{y} = \operatorname*{argmax}_{\mathbf{y} \in \Upsilon^*} \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$$

(DP only works if Φ has 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

Column-Generation Algorithm

1. $\Upsilon_n^1 = \emptyset$, for $n = 1, \ldots, N$
2. Solve

$$(\mathbf{w}^t, \rho, \boldsymbol{\xi}^t) = \underset{\|\mathbf{w}\|=1, \rho, \boldsymbol{\xi}}{\operatorname{argmax}} \quad \rho - C \sum_{n=1}^{N} \xi_n$$

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

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

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

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

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

Problems

- Optimization may require many iterations
- Number of variables increases linearly
- When using kernels, solving optimization problems can become easily 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 needed
- Idea: Decompose problem
    - First part uses kernels, can be precomputed
    - Second part without kernels and only combines ingredients

$\Rightarrow$ Solve problems with $10,000$ examples, instead of just ≤ 500

Problems

- Optimization may require many iterations
- Number of variables increases linearly
- When using kernels, solving optimization problems can become easily 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 needed
- Idea: Decompose problem
  - First part uses kernels, can be precomputed
  - Second part without kernels and only combines ingredients

$\Rightarrow$ Solve problems with $10,000$ examples, instead of just ≤ 500

Problems

- Optimization may require many iterations
- Number of variables increases linearly
- When using kernels, solving optimization problems can become easily 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 needed
- Idea: Decompose problem
  - First part uses kernels, can be precomputed
  - Second part without kernels and only combines ingredients

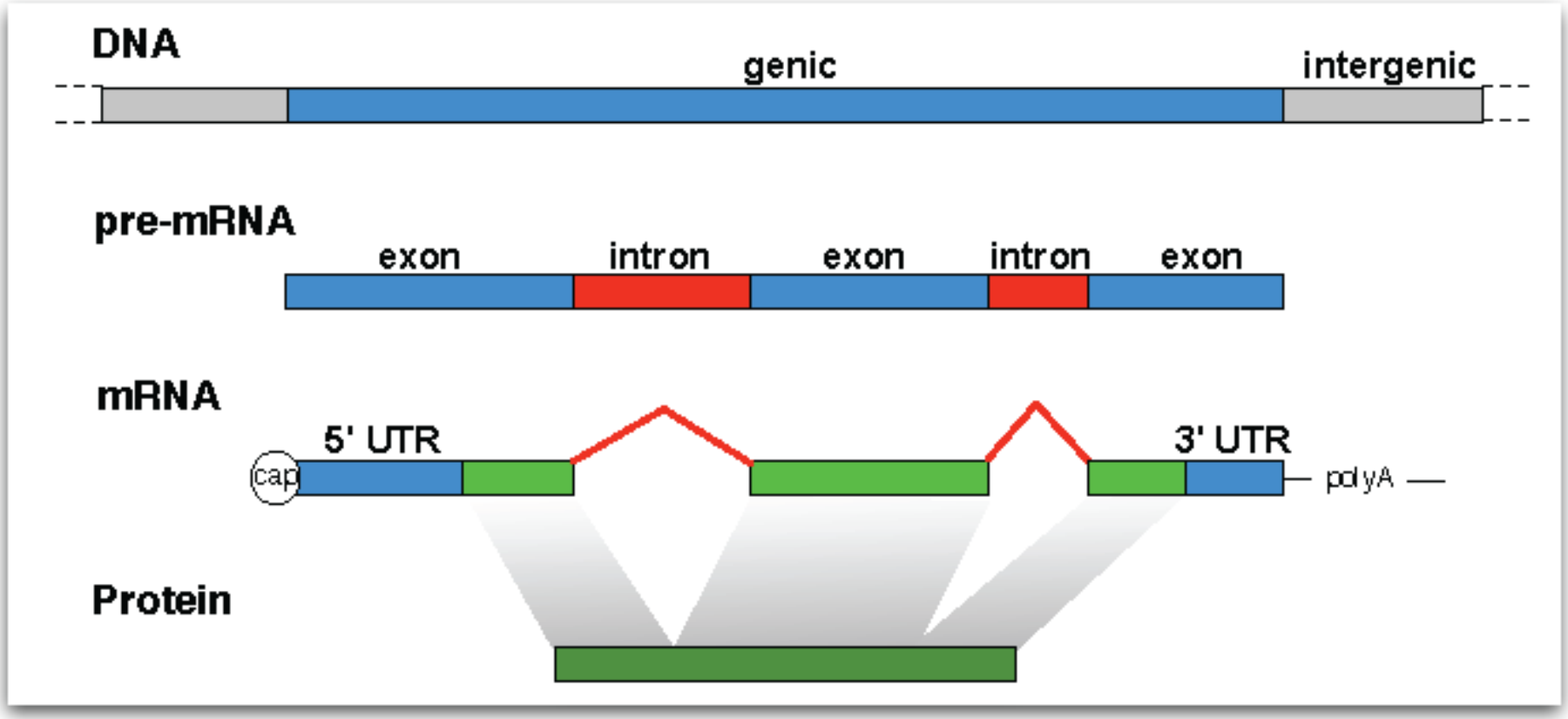
$\Rightarrow$ Solve problems with $10,000$ examples, instead of just ≤ 500

Problems

- Optimization may require many iterations
- Number of variables increases linearly
- When using kernels, solving optimization problems can become easily 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 needed
- Idea: Decompose problem
  - First part uses kernels, can be precomputed
  - Second part without kernels and only combines ingredients

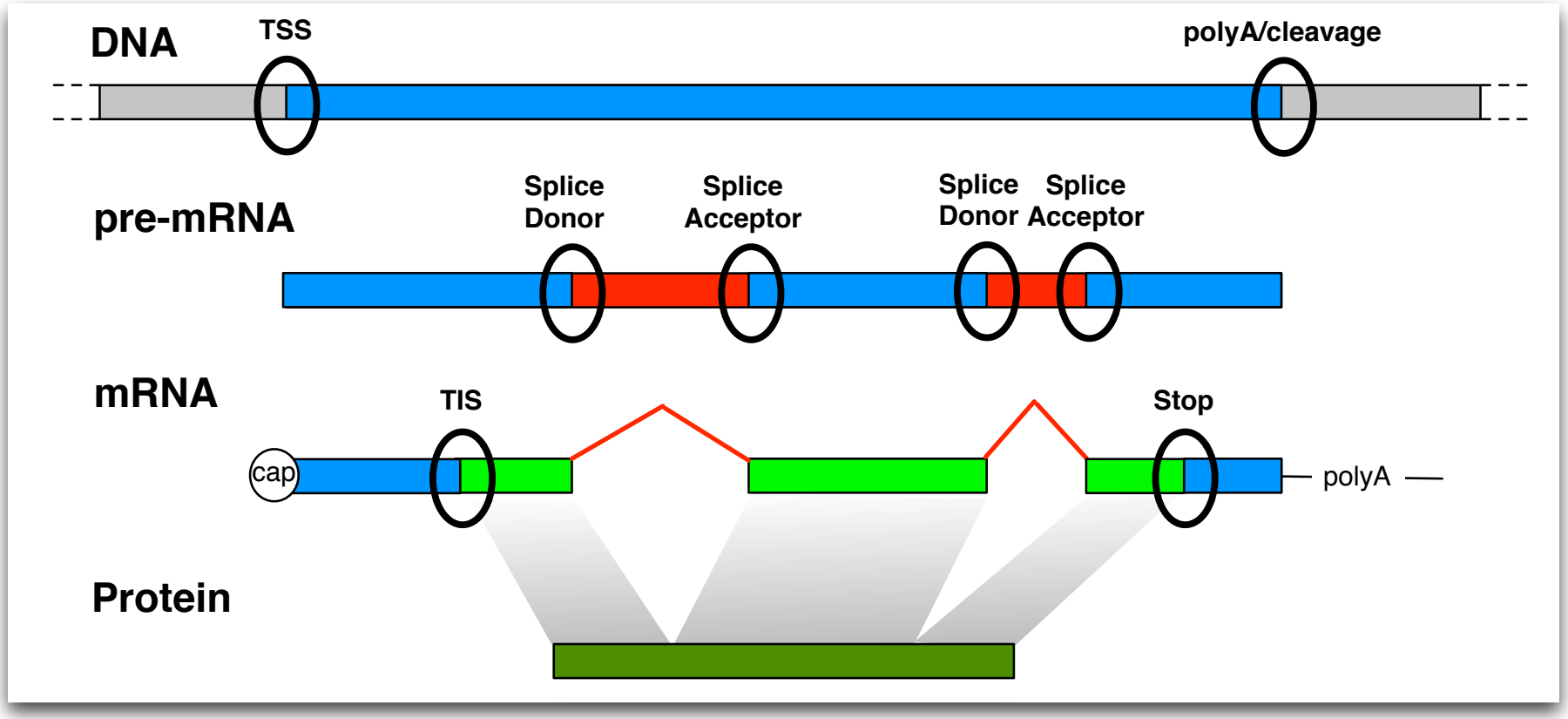
$\Rightarrow$ Solve problems with $10,000$ examples, instead of just ≤ 500

Back to Computational Gene Finding



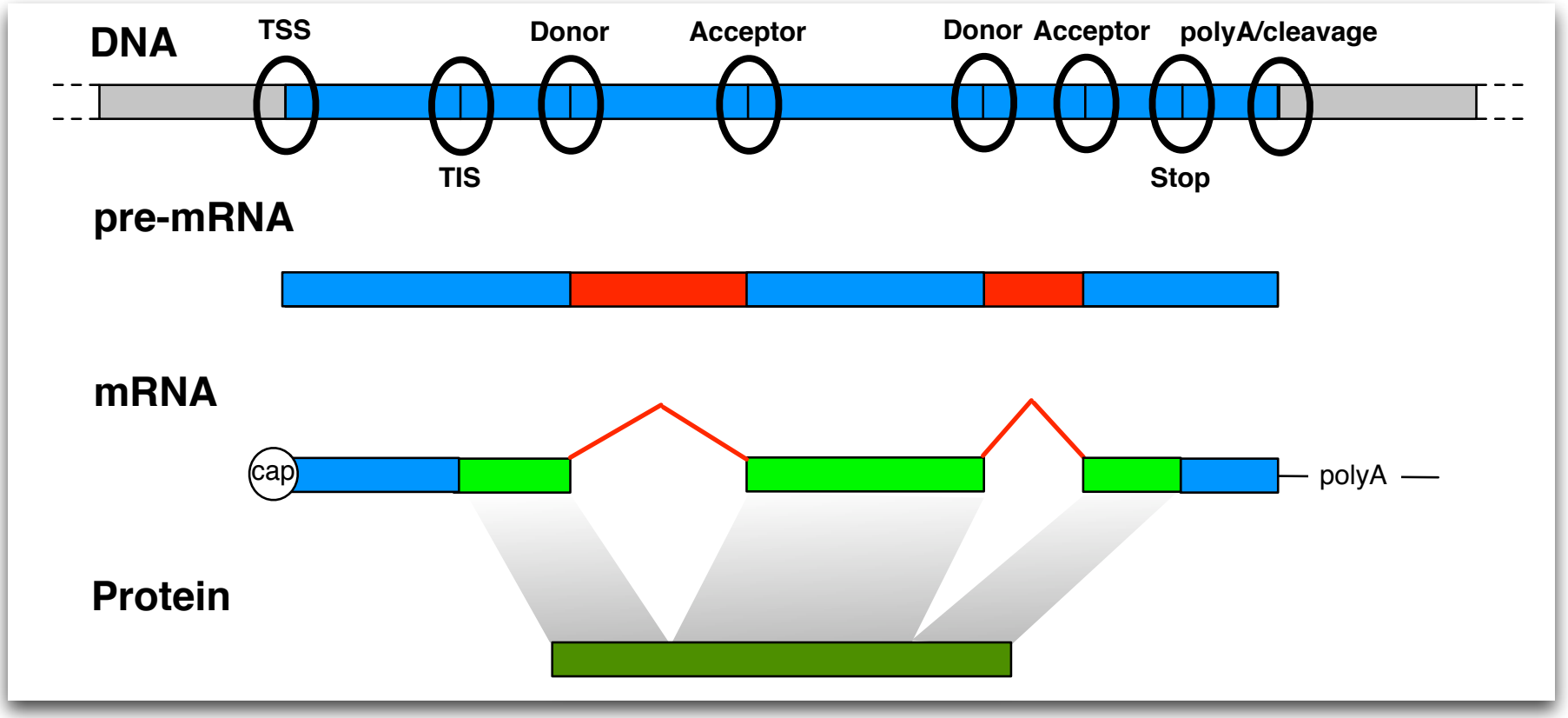
- Given a piece of DNA sequence
- Predict gene products inluding the intermediate processing steps

Back to Computational Gene Finding



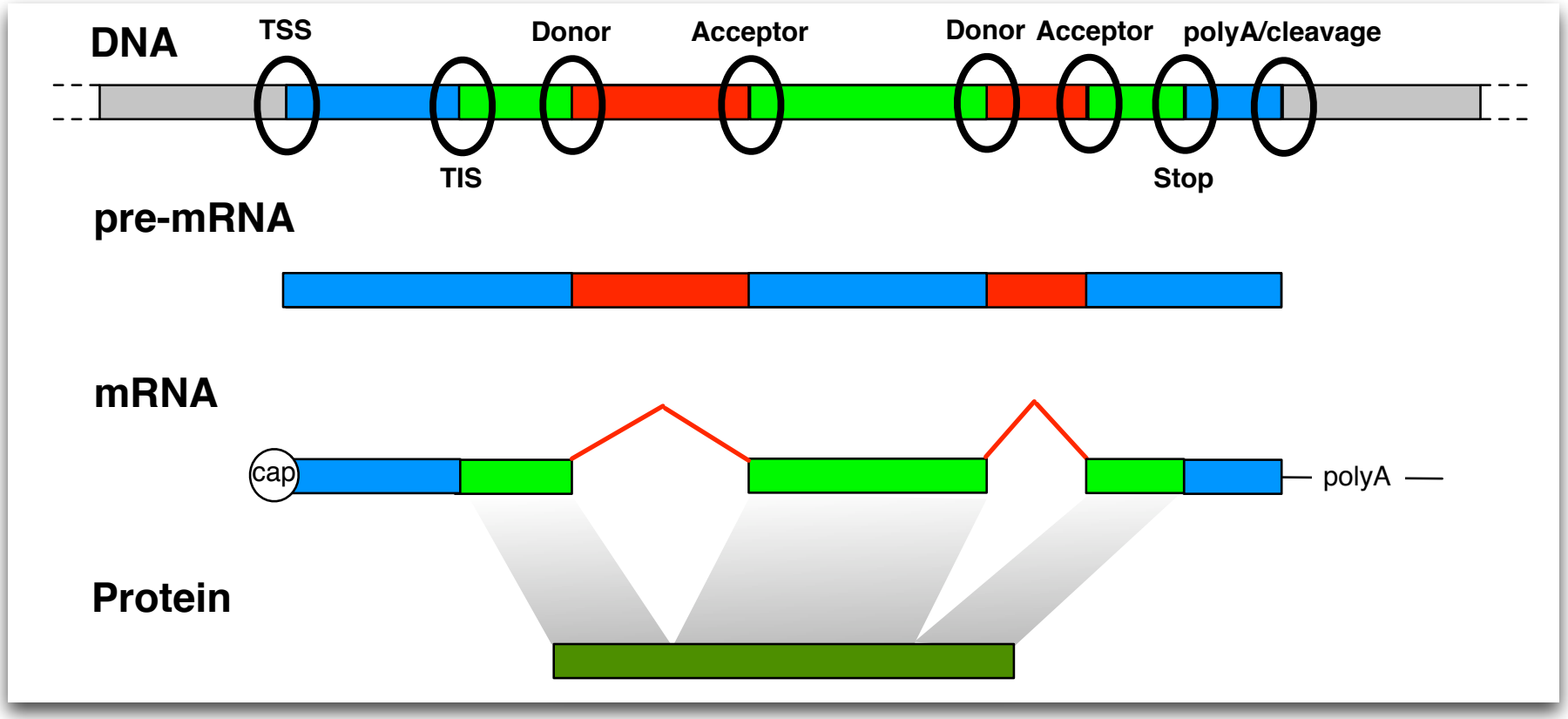
- Predict signals used during processing

Back to Computational Gene Finding



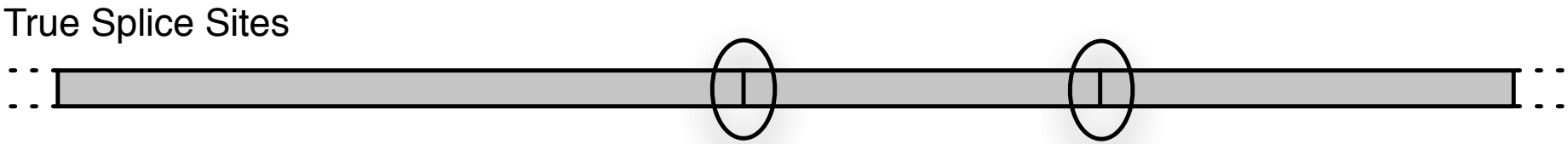
- Predict signals used during processing

Back to Computational Gene Finding

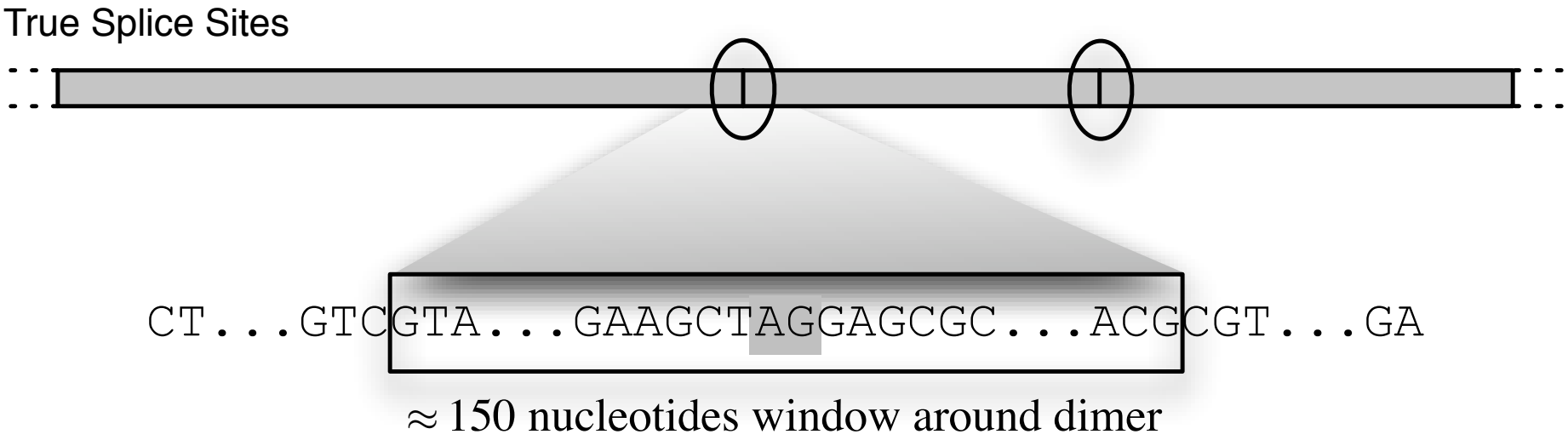


- Predict signals used during processing
- Predict the correct corresponding **label sequence** with labels "intergenic", "exon", "intron", "5' UTR", etc.

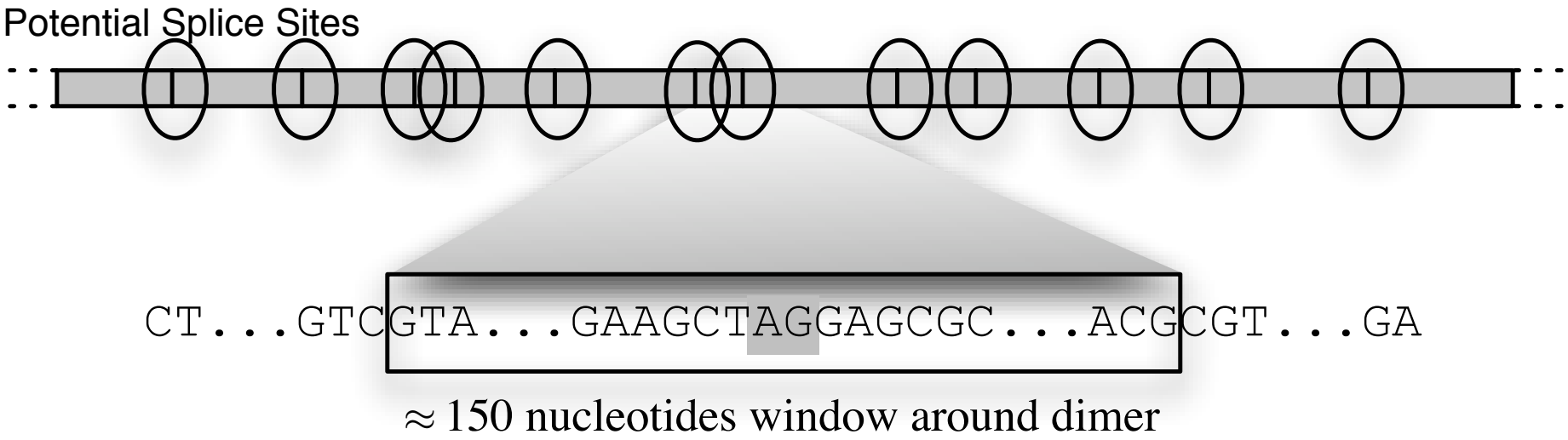
Example: Splice Site Recognition



Example: Splice Site Recognition

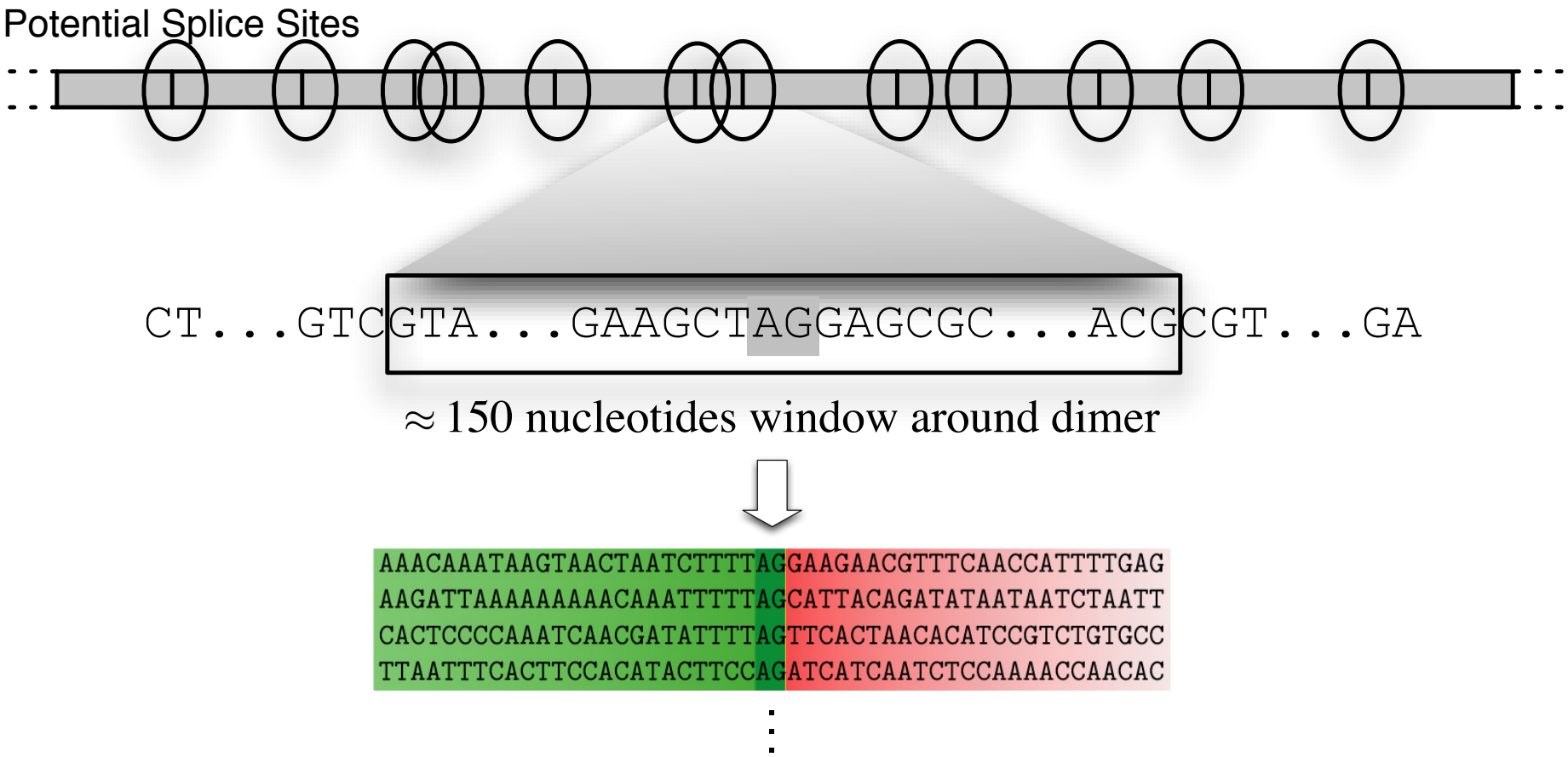


Example: Splice Site Recognition



≈ 150 nucleotides window around dimer

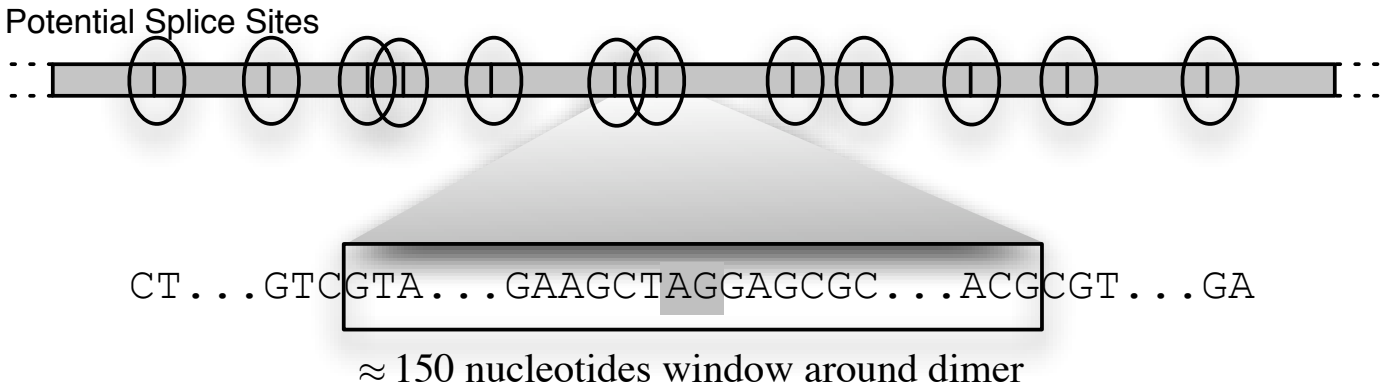
Example: Splice Site Recognition



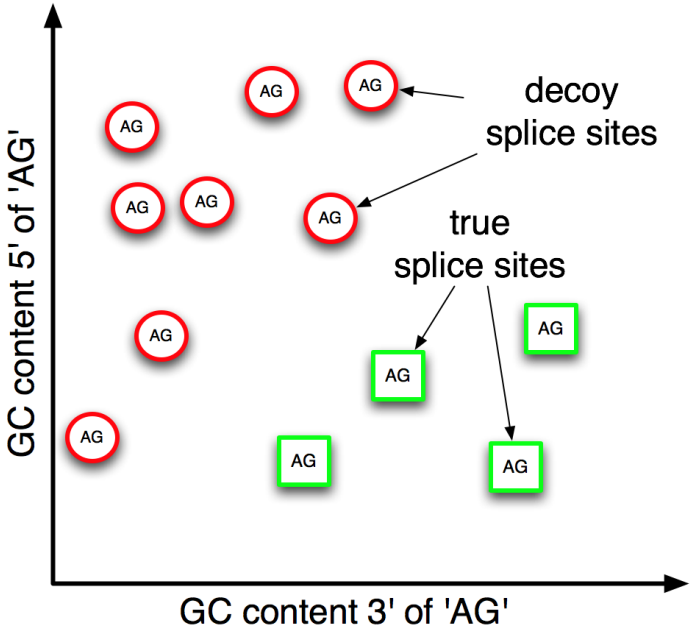
- True sites: fixed window around a true splice site
- Decoy sites: all other consensus sites

$\Rightarrow$ **Millions of examples from EST databases**

Example: Splice Site Recognition



Basic idea:



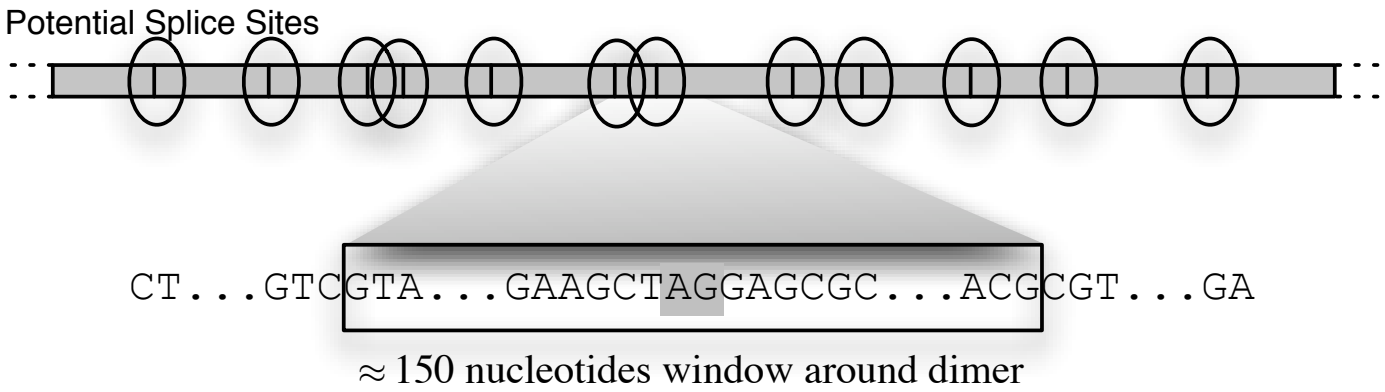
For instance, exploit that exons have higher GC content

or

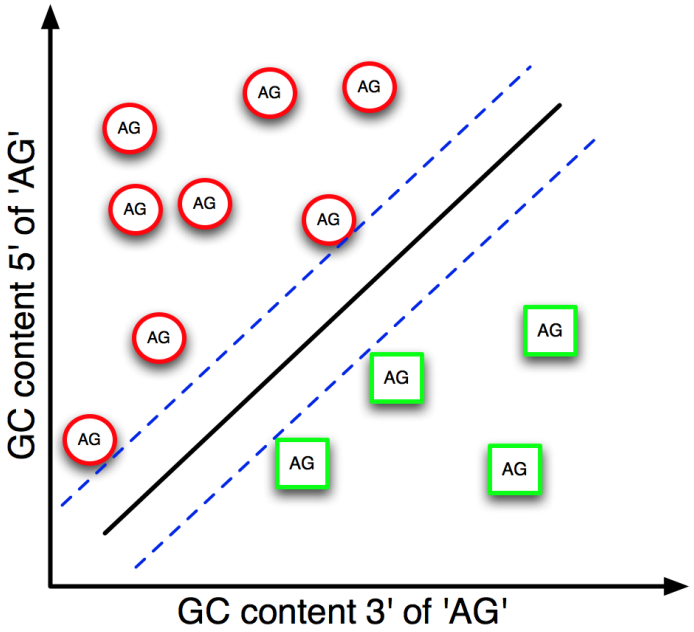
that specific motifs appear near splice sites.

[Sonnenburg et al., 2007b]

Example: Splice Site Recognition



Basic idea:

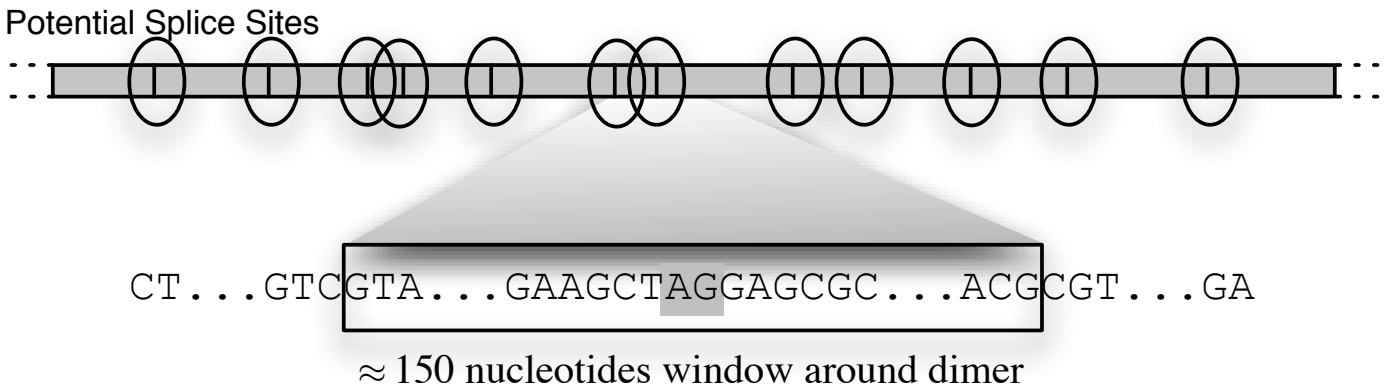


In practice: Use one feature per possible substring (e.g. $\leq$20) at all positions

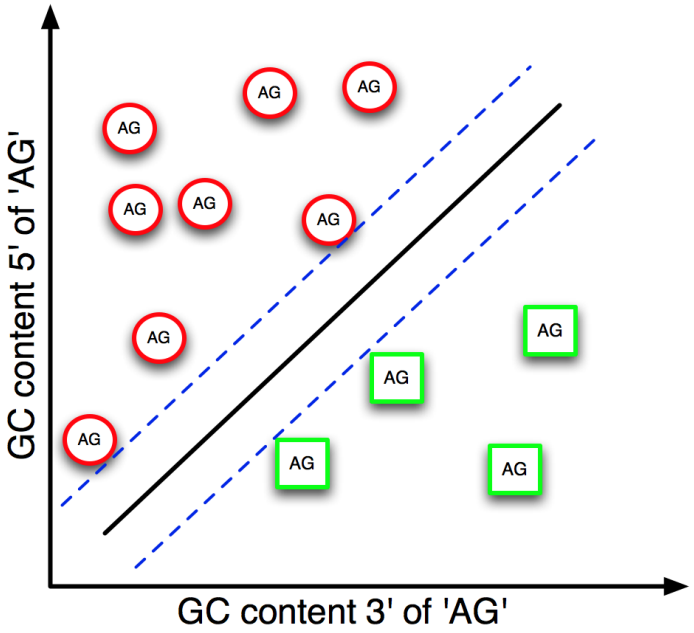
$150 \cdot (4^1 + \ldots + 4^{20}) \approx 2 \cdot 10^{14}$ features

- Needs efficient algorithms
- Leads to most accurate predictors that currently exist

Example: Splice Site Recognition



Basic idea:

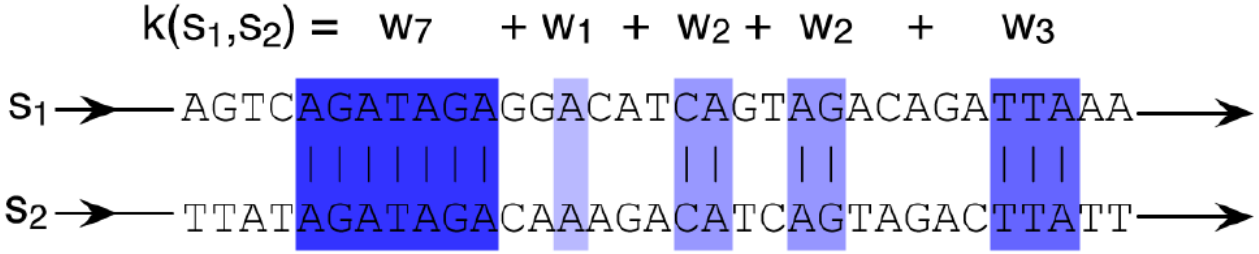


In practice: Use one feature per possible substring (e.g. $\leq$20) at all positions

$150 \cdot (4^1 + \ldots + 4^{20}) \approx 2 \cdot 10^{14}$ features

- Needs efficient algorithms
- Leads to most accurate predictors that currently exist

Good news: Use **WD kernel** to implicitly consider features:



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

Drawback: Training too expensive for millions of examples

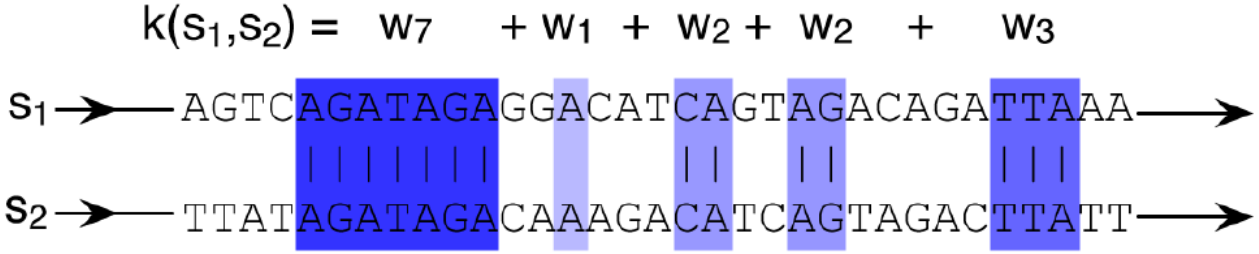
Needs more sophisticated algorithms:

- Exploit that features are sparse and can be explicitly represented
  - String indexing data structures [Sonnenburg et al., 2007a]
- Use novel optimization techniques
  - Use subgradient-based optimization [Franc and Sonnenburg, 2009]

Implemented in a freely available software package

www.shogun-toolbox.org [Sonnenburg et al., 2007a]

Good news: Use **WD kernel** to implicitly consider features:



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

Drawback: Training is too expensive for millions of examples

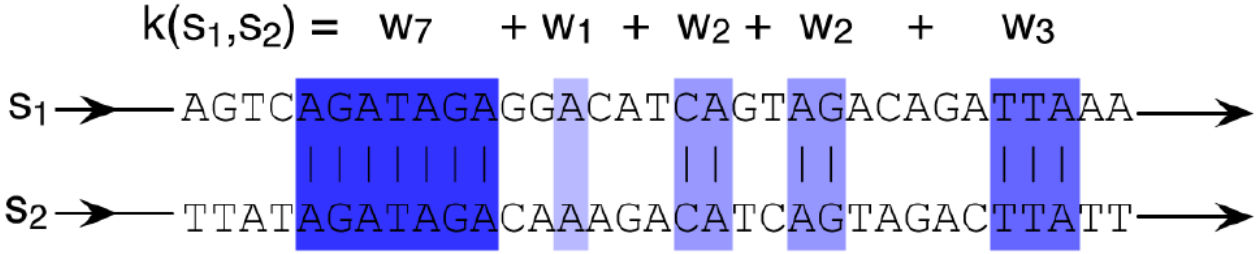
Needs more sophisticated algorithms:

- Exploit that features are sparse and can be explicitly represented
  - String indexing data structures [Sonnenburg et al., 2007a]
- Use novel optimization techniques
  - Use subgradient-based optimization [Franc and Sonnenburg, 2009]

Implemented in a freely available software package

www.shogun-toolbox.org [Sonnenburg et al., 2007a]

Good news: Use **WD kernel** to implicitly consider features:



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

Drawback: Training was too expensive for millions of examples

Needs more sophisticated algorithms:

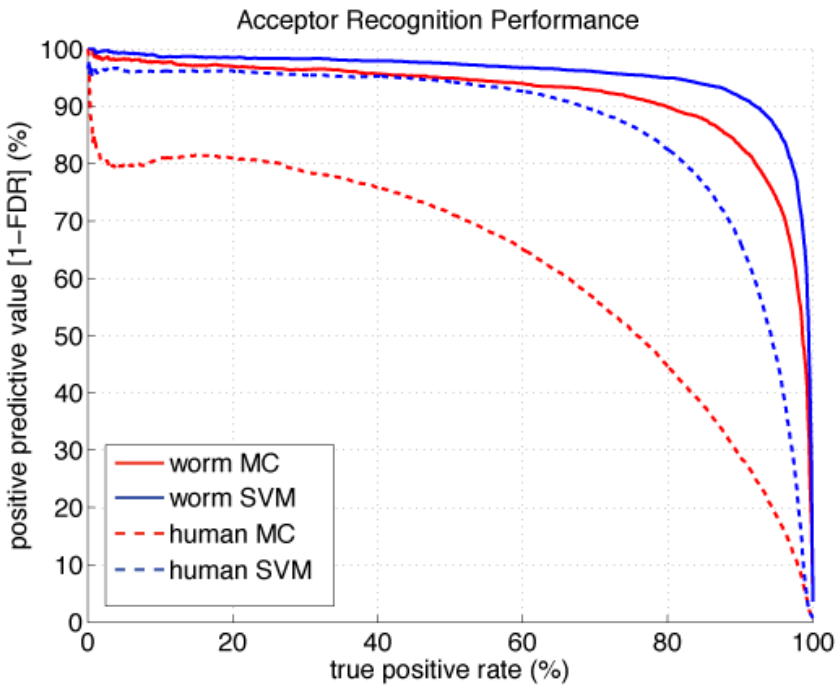
- Exploit that features are sparse and can be explicitly represented
  - String indexing data structures [Sonnenburg et al., 2007a]
- Use novel optimization techniques
  - Use subgradient-based optimization [Franc and Sonnenburg, 2009]

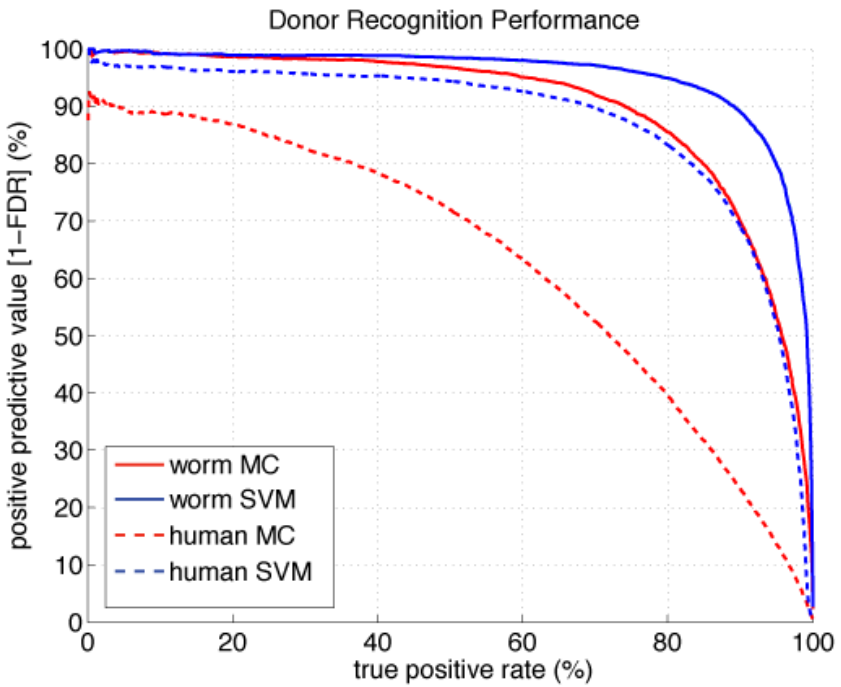
Implemented in a freely available software package

www.shogun-toolbox.org [Sonnenburg et al., 2007a]

Results on Splice Site Recognition

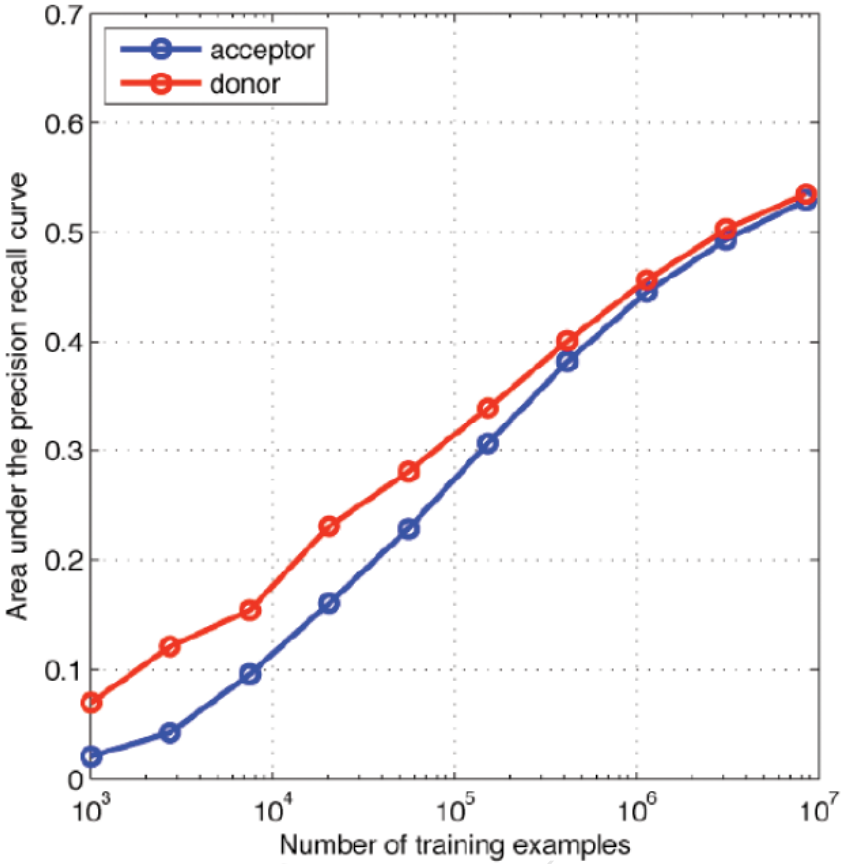
	Worm		Fly		Cress		Fish		Human	
	Acc	**Don**	**Acc**	**Don**	**Acc**	**Don**	**Acc**	**Don**	**Acc**	**Don**
Markov Chain auPRC(%)	92.1	90.0	80.3	78.5	87.4	88.2	63.6	62.9	16.2	26.0
SVM auPRC(%)	95.9	95.3	86.7	87.5	92.2	92.9	86.6	86.9	54.4	56.9

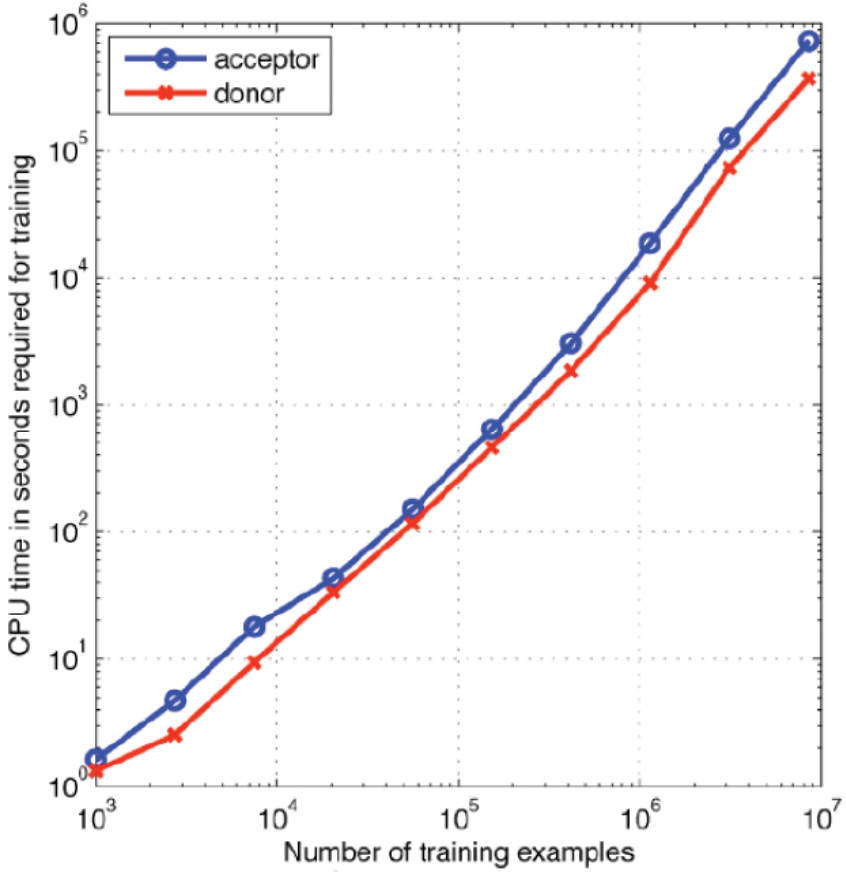




[Sonnenburg, Schweikert, Philips, Behr, Rätsch, 2007]

Human Splice Sites with WD Kernel



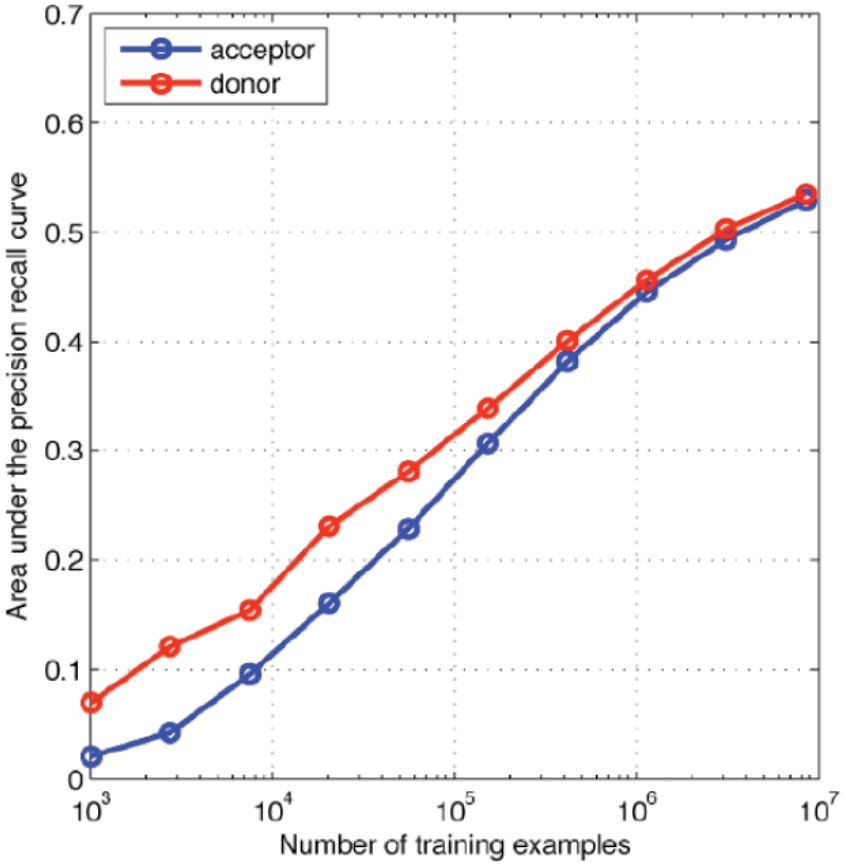


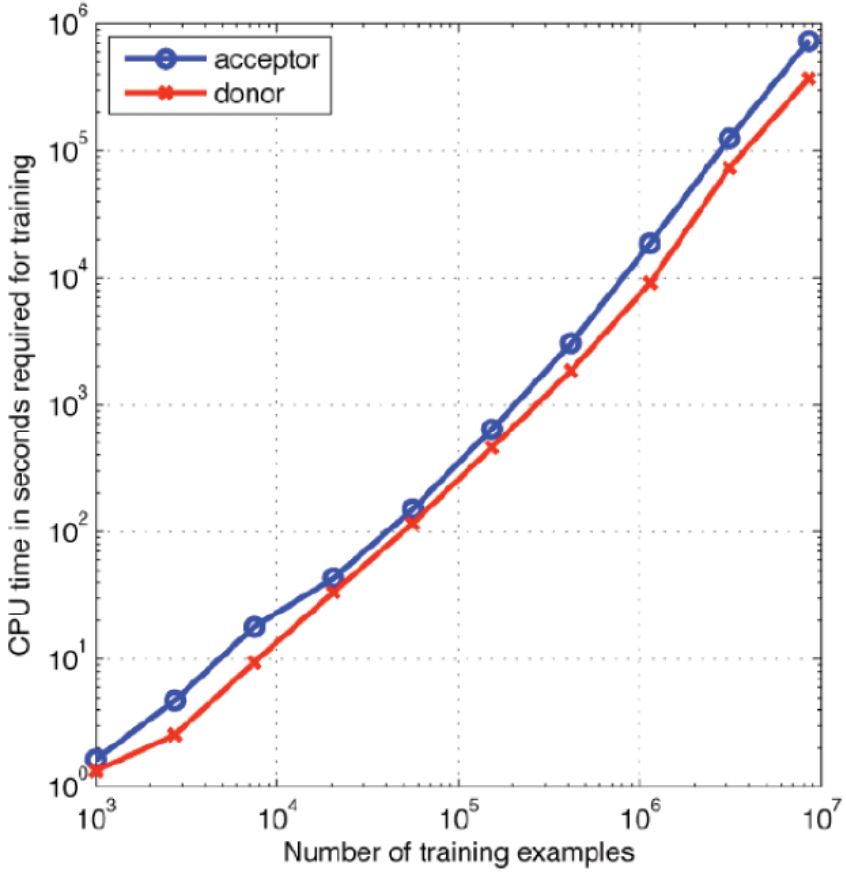
Performance (=Area under PRC curve) keeps increasing.

⇒ **Pays off to use all available data!**

[Sonnenburg et al., 2007b]

Human Splice Sites with WD Kernel



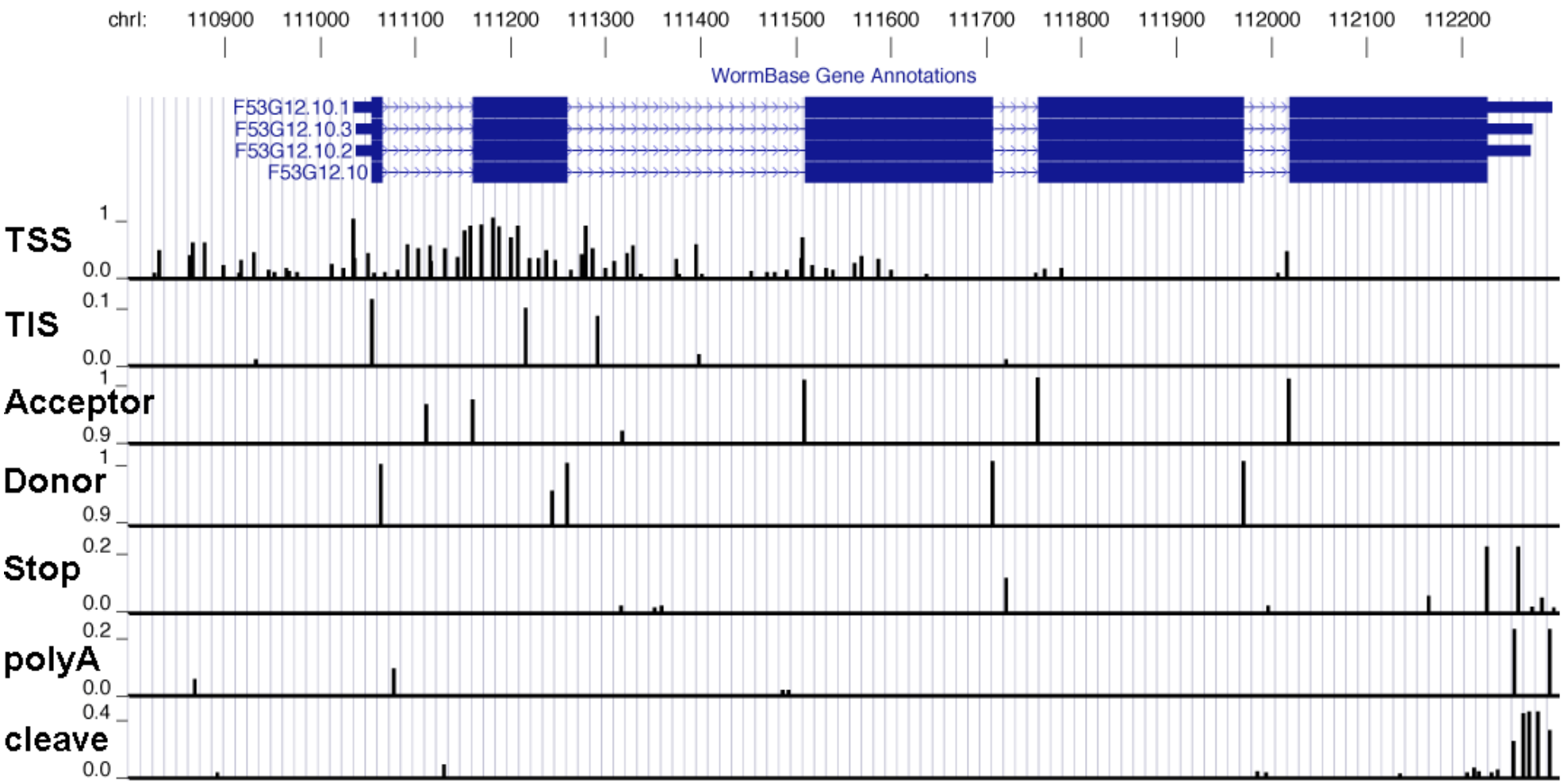


Performance (=Area under PRC curve) keeps increasing.

$\Rightarrow$ **Pays off to use all available data!**

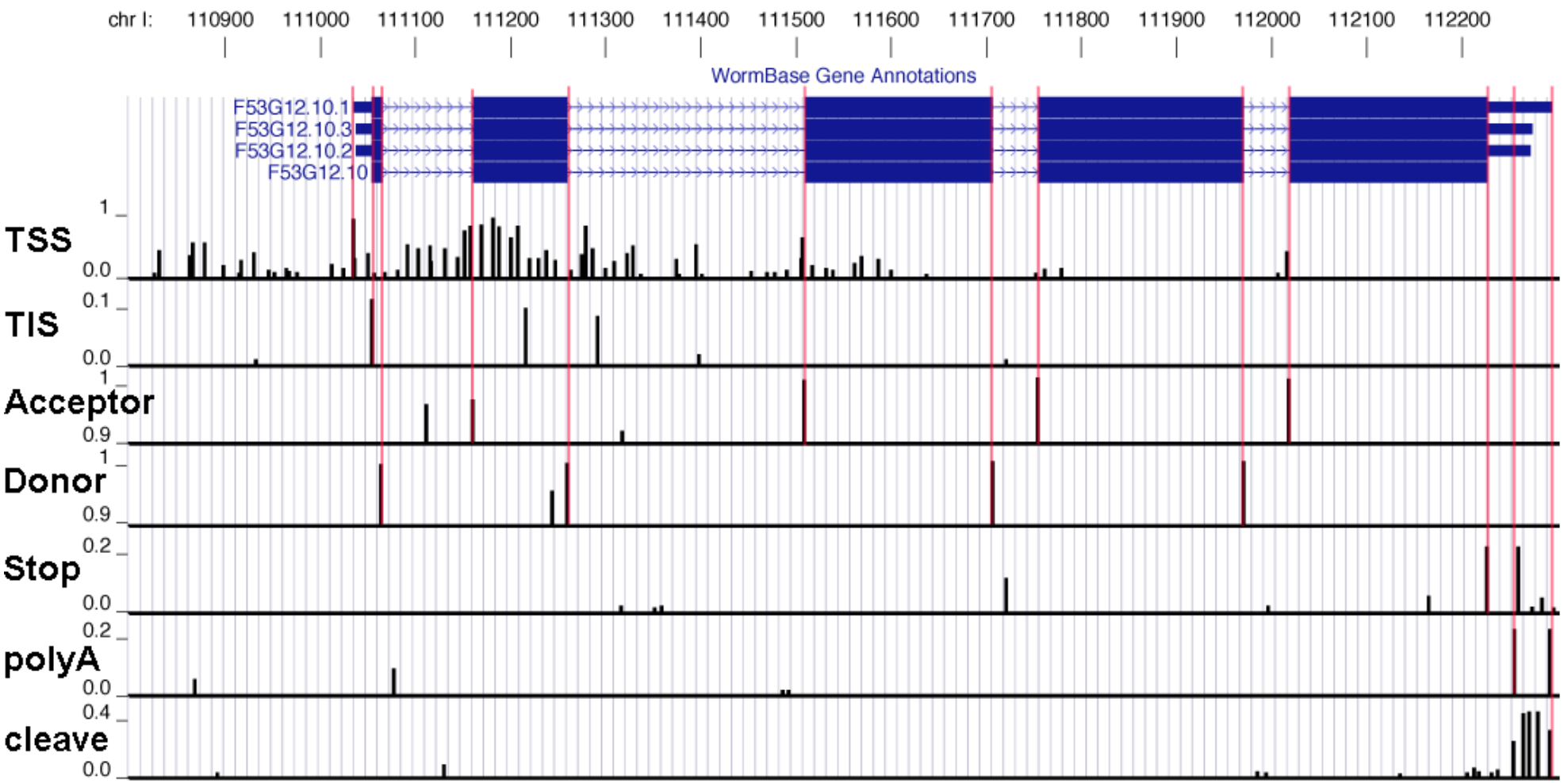
[Sonnenburg et al., 2007b]

Example: Predictions in UCSC Browser



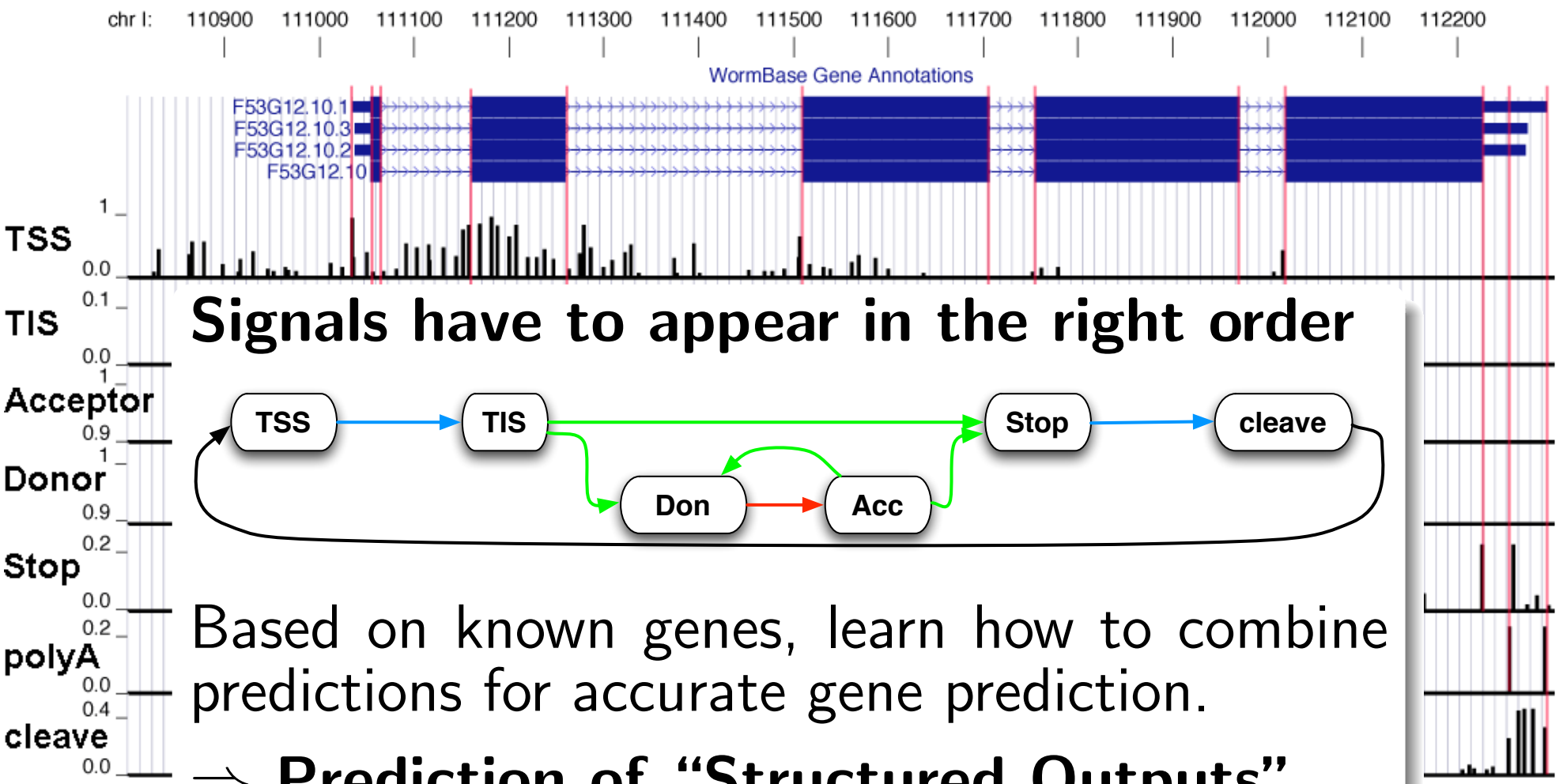
[Rätsch et al., 2007, Schweikert et al., 2009c]

Example: Predictions in UCSC Browser



[Rätsch et al., 2007, Schweikert et al., 2009c]

Example: Predictions in UCSC Browser



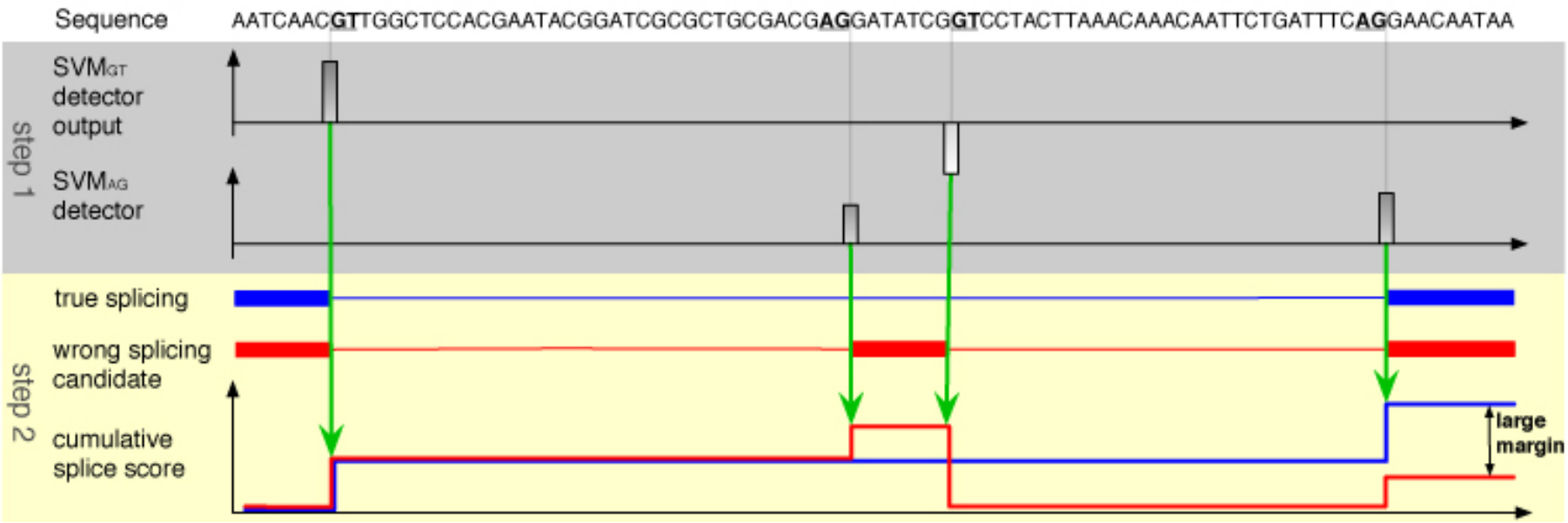
Signals have to appear in the right order

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

$\Rightarrow$ **Prediction of "Structured Outputs"**

[Rätsch et al., 2007, Schweikert et al., 2009c]

Discriminative Gene Prediction (simplified)



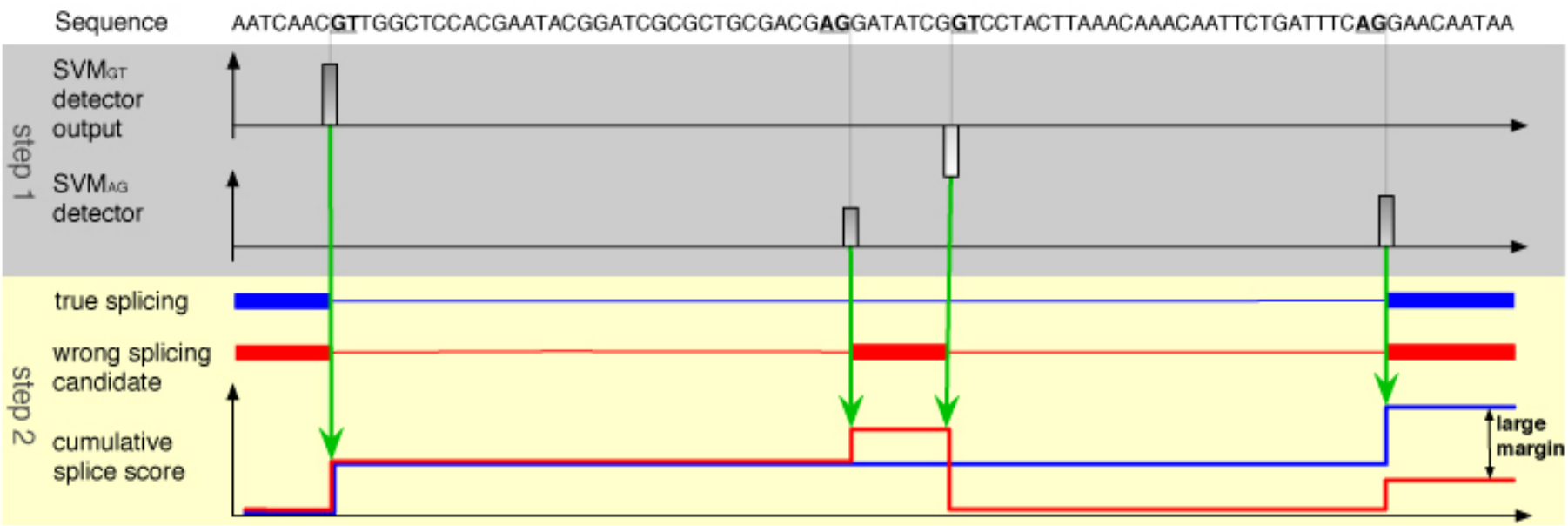
[Rätsch, Sonnenburg, Srinivasan, Witte, Müller, Sommer, Schölkopf, 2007]

Simplified Model: Score for splice form $\mathbf{y} = \{(p_j, q_j)\}_{j=1}^{J}$:

$$F(\mathbf{y}) := \underbrace{\sum_{j=1}^{J-1} S_{GT}(f_j^{GT}) + \sum_{j=2}^{J} S_{AG}(f_j^{AG})}_{\text{Splice signals}} + \underbrace{\sum_{j=1}^{J-1} S_{L_I}(p_{j+1} - q_j) + \sum_{j=1}^{J} S_{L_E}(q_j - p_j)}_{\text{Segment lengths}}$$

Tune free parameters by solving **optimization problem**.

Discriminative Gene Prediction (simplified)



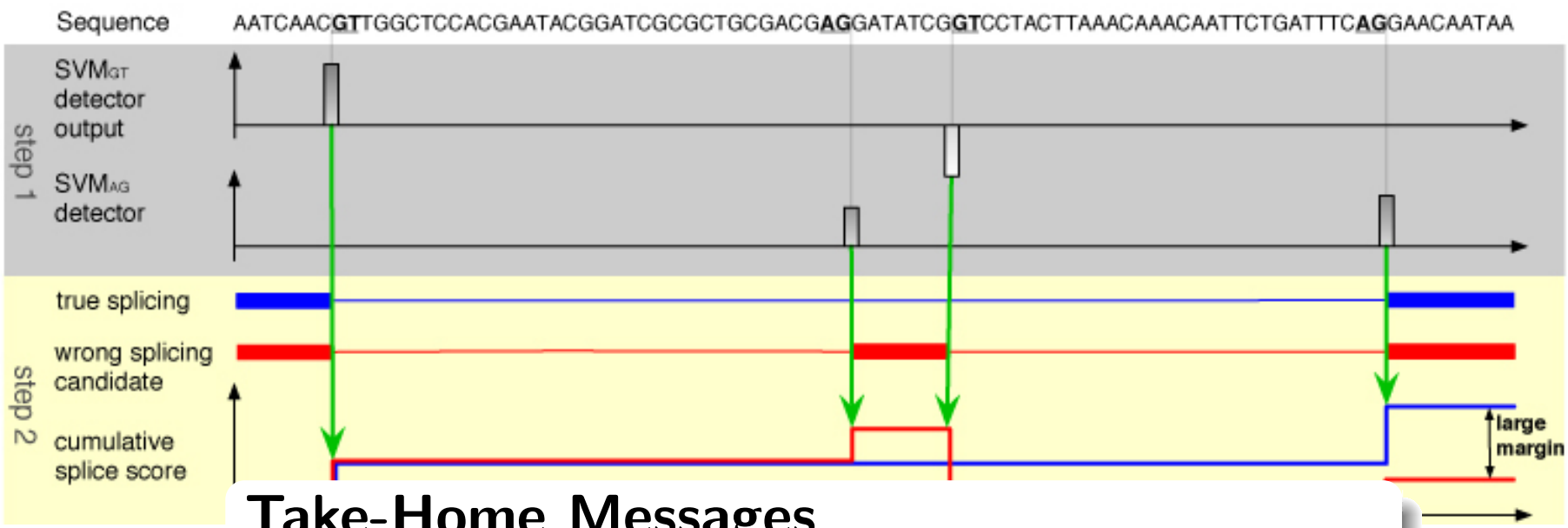
[Rätsch, Sonnenburg, Srinivasan, Witte, Müller, Sommer, Schölkopf, 2007]

Simplified Model: Score for splice form $\mathbf{y} = \{(p_j, q_j)\}_{j=1}^{J}$:

$$F(\mathbf{y}) := \underbrace{\sum_{j=1}^{J-1} S_{GT}(f_j^{GT}) + \sum_{j=2}^{J} S_{AG}(f_j^{AG})}_{\text{Splice signals}} + \underbrace{\sum_{j=1}^{J-1} S_{L_I}(p_{j+1} - q_j) + \sum_{j=1}^{J} S_{L_E}(q_j - p_j)}_{\text{Segment lengths}}$$

Tune free parameters by solving **optimization problem**.

Discriminative Gene Prediction (simplified)



[Schölkopf, 2007]

Simplified

$$F(\mathbf{y}) := \underbrace{\sum_{j=1}^{J-1}}$$

$p_j)$

Tune free

Take-Home Messages

- **Avoid solving a huge optimization problem by splitting it into two parts**
- **Exploit state-of-the-art signal predictions**
- **Predict more accurately than generative models**

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., 2009c]

	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., 2009c]

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., 2009b]

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., 2009c]

	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., 2009c]

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., 2009b]

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., 2009c]

	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., 2009c]

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., 2009b]

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., 2009c]

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

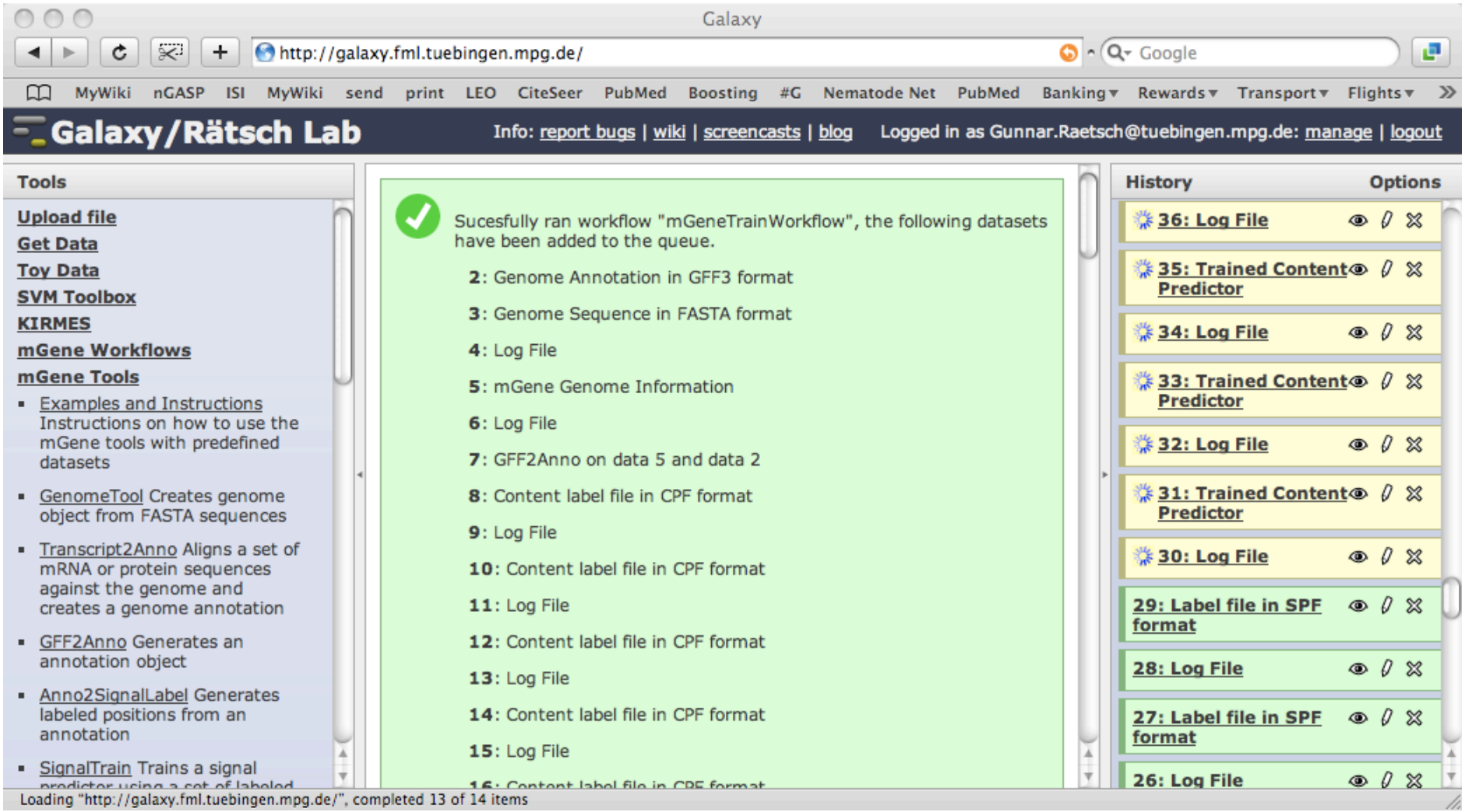
- Annotation of other nematode genomes: [Schweikert et al., 2009c]

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., 2009b]

mGene.web: Gene Finding for Everybody ;-)

Friedrich Miescher Laboratory
of the Max Planck Society



mgene.org/web

[Schweikert et al., 2009b]

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"

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

[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

[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

[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

[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%**

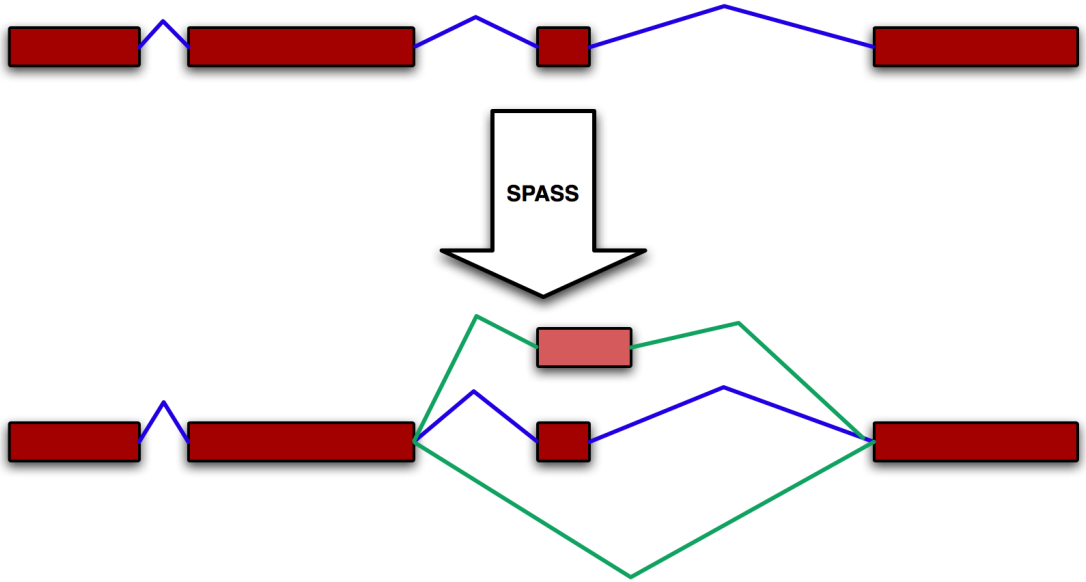
In progress:

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

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

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: [Zien et al., 2006]

    Input: DNA sequence
    Output: Splicing graph

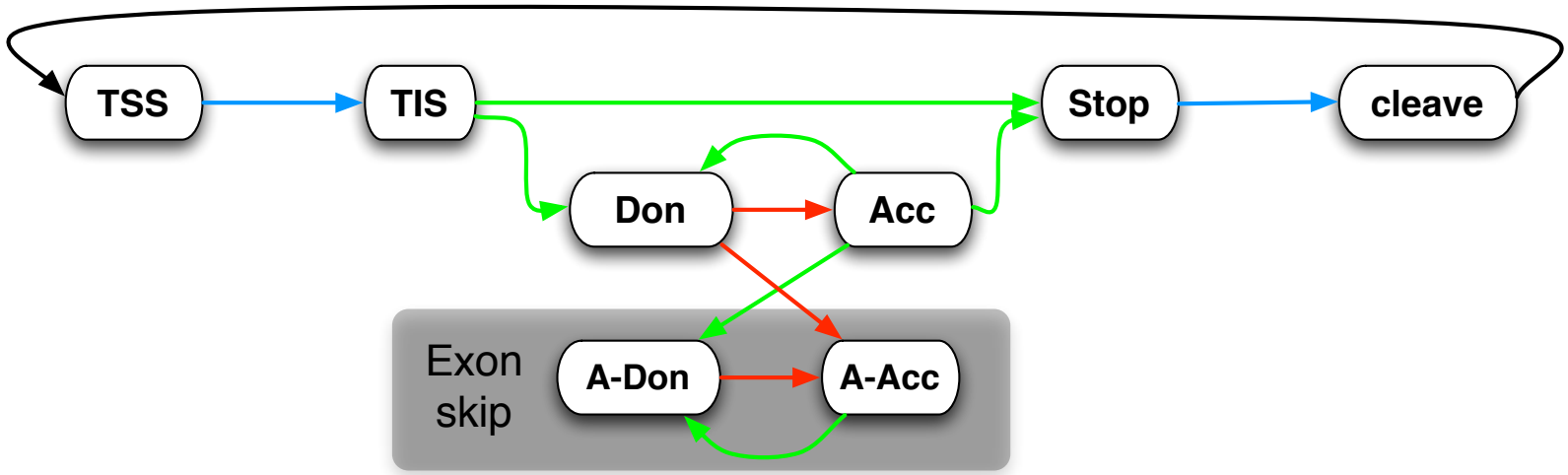
- mGene approach can be extended to
  - Predict **"simple splicing graphs"**

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

Alternative Transcripts: More Steps

- Combine gene finding with prediction of alternative splicing
  - Machine learning challenge: [Zien et al., 2006]

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

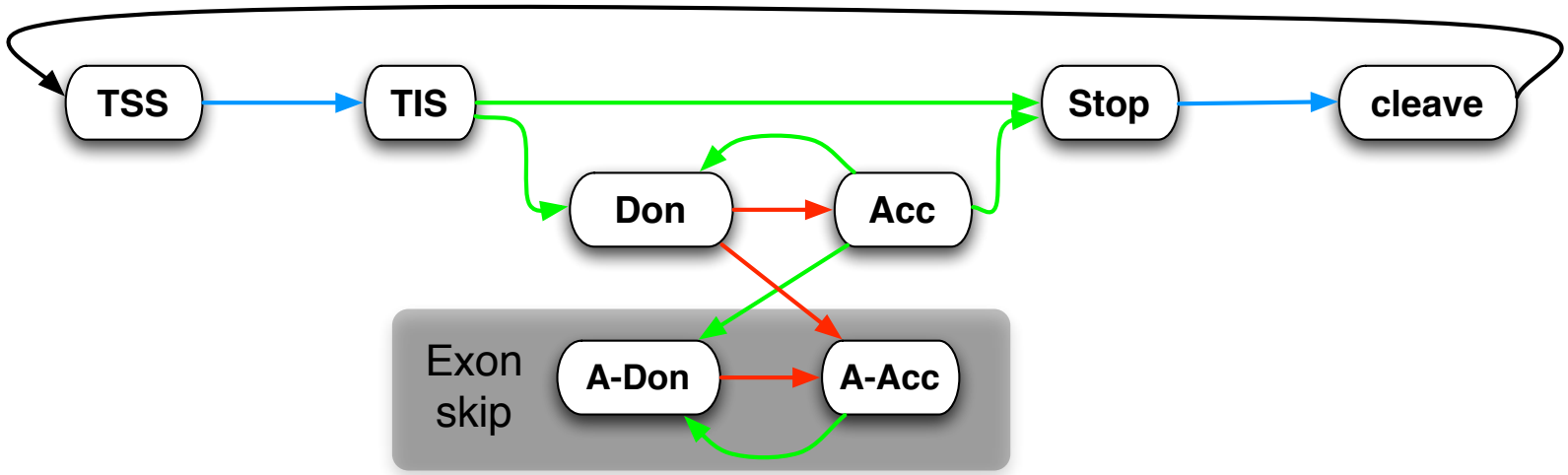


- But, without **clean data** we cannot train the model ...

Alternative Transcripts: More Steps

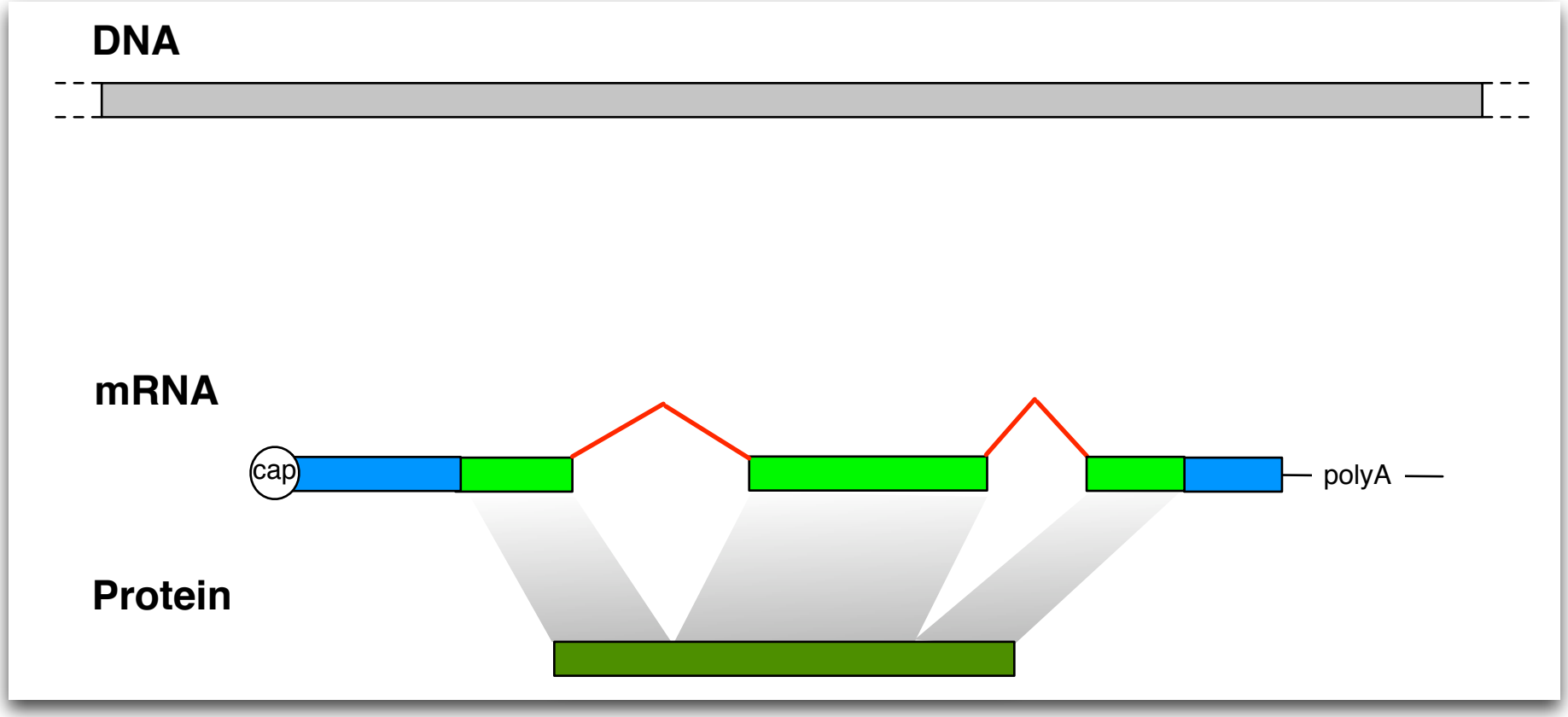
- Combine gene finding with prediction of alternative splicing
  - Machine learning challenge: [Zien et al., 2006]

    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 ...

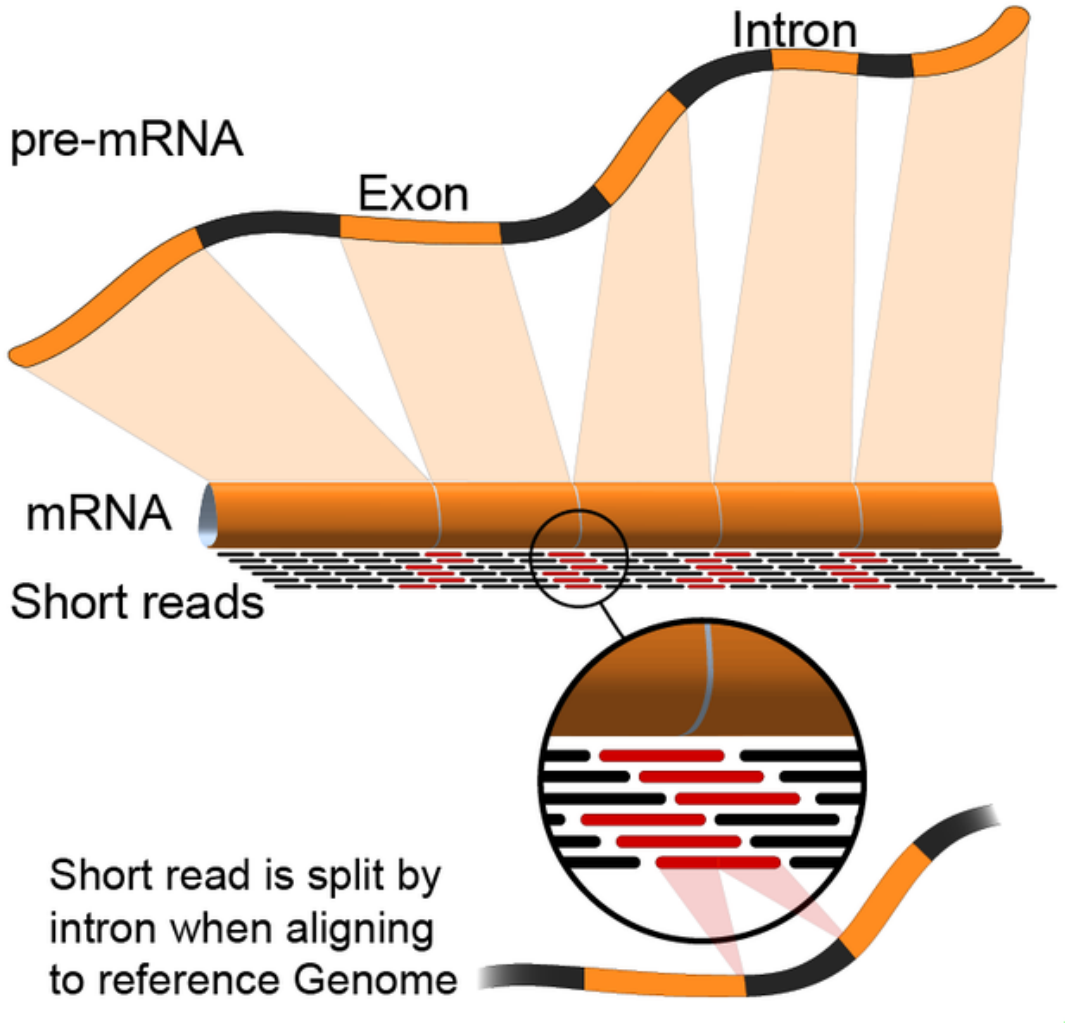
Reconstructing the Transcriptome



Directly measure the transcriptome

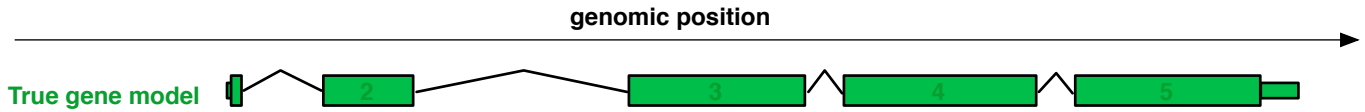
- Whole genome tiling arrays
- **Transcriptome sequencing** (Sanger or Next Generation Sequencing)

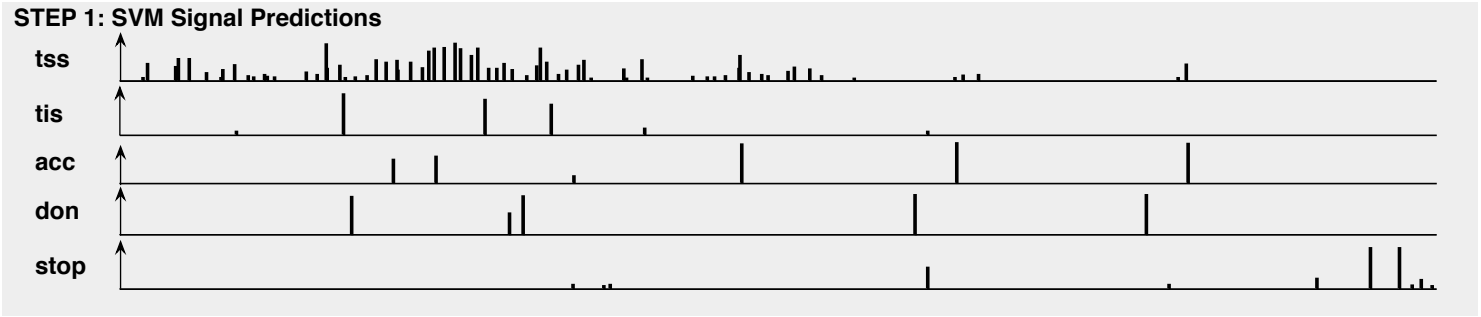
mRNA Deep Sequencing

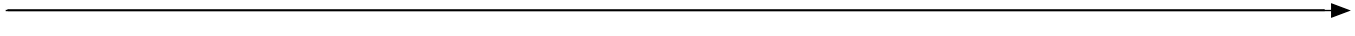


[Wikipedia]

Learning to Integrate Data Sources

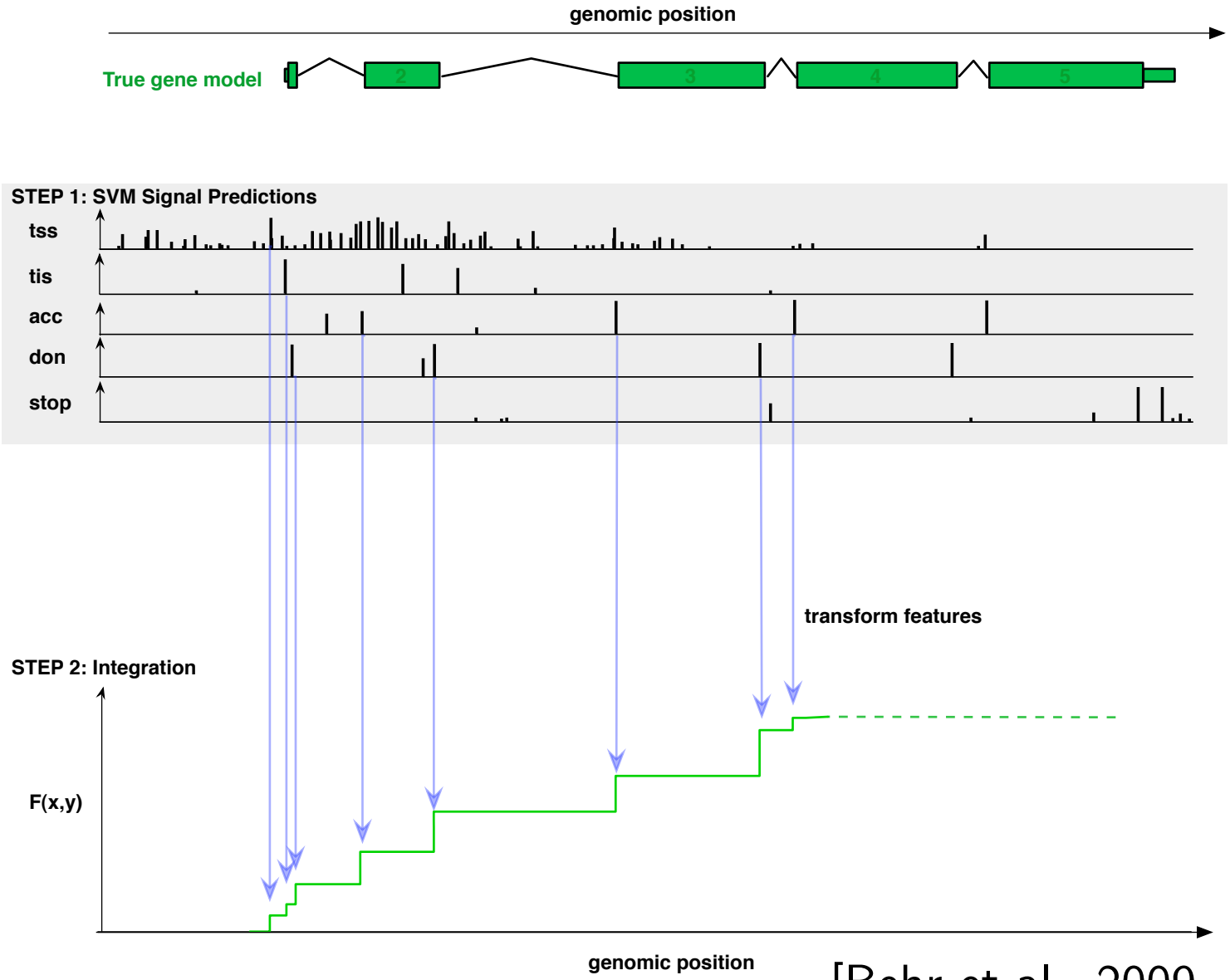






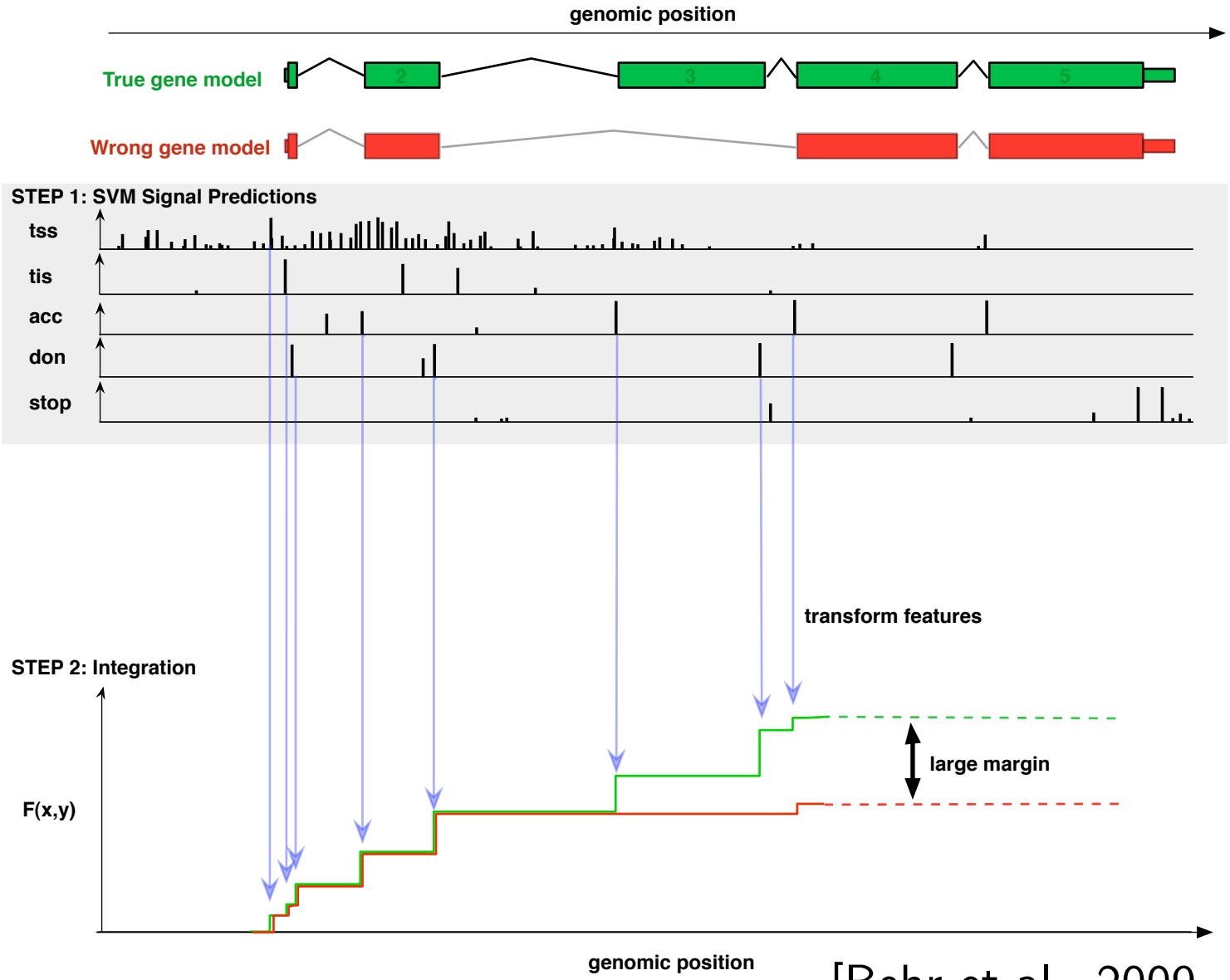
[Behr et al., 2009, in prep.]

Learning to Integrate Data Sources



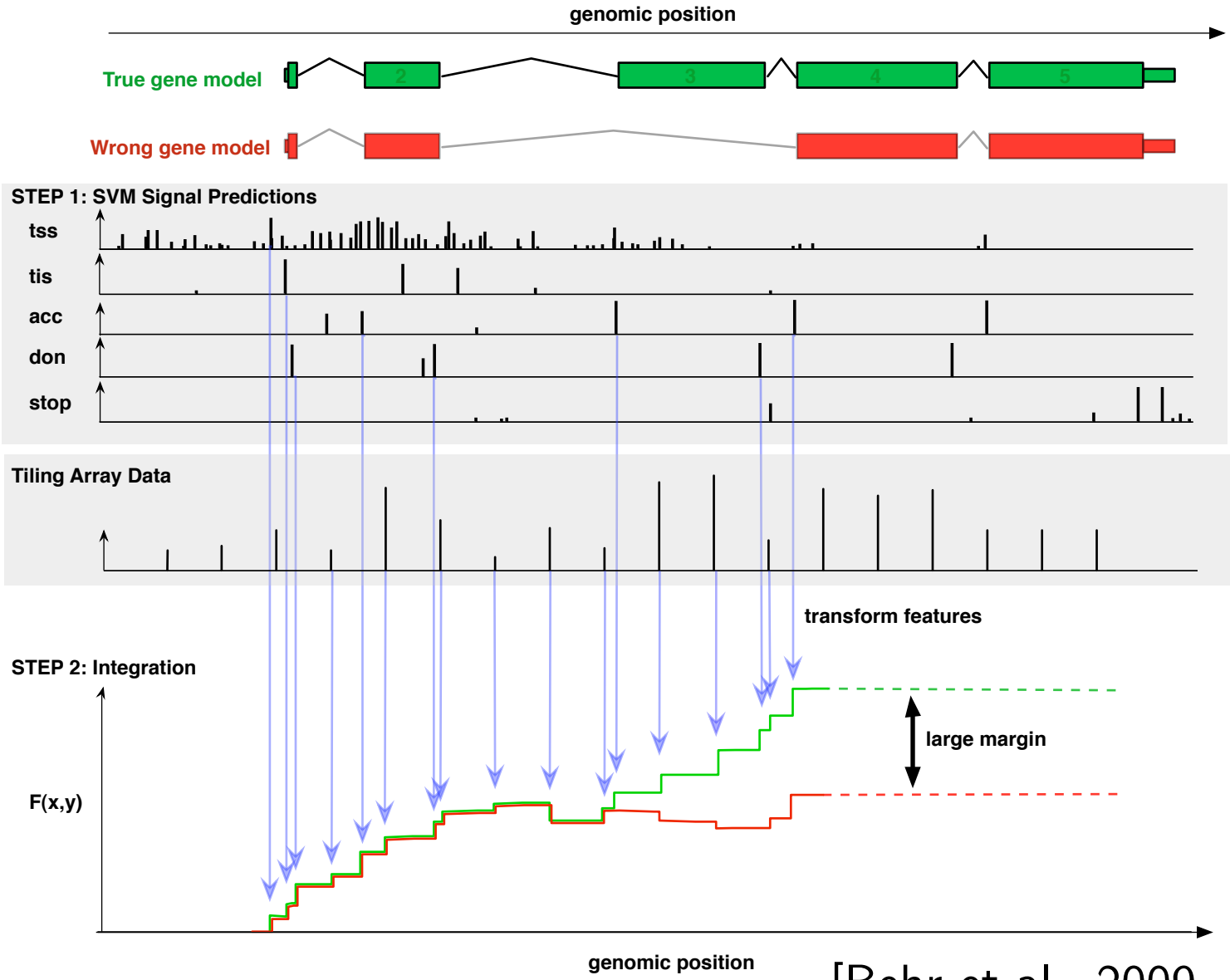
[Behr et al., 2009, in prep.]

Learning to Integrate Data Sources



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

Learning to Integrate Data Sources



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

Idea: Use transcriptome measurements to enhance recognition of exonic regions.

Results for A. thaliana: (Comparison with known gene models)

	transcript level $(SN + SP)/2$
1. mGene (*ab initio*) ...	**73.3%**
2. ... with tiling arrays (11 tissues)	**82.1%**
3. ... with mRNA-seq (1 tissue)	**81.1%**

Transcriptome measurements certainly help, but predictions are still quite far from perfect.

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

Idea: Use transcriptome measurements to enhance recognition of exonic regions.

Results for A. thaliana: (Comparison with known gene models)

	transcript level $(SN + SP)/2$
1. mGene (*ab initio*) ...	**73.3%**
2. ...with tiling arrays (11 tissues)	**82.1%**
3. ...with mRNA-seq (1 tissue)	**81.1%**

Transcriptome measurements certainly help, but predictions are still quite far from perfect.

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

Idea: Use transcriptome measurements to enhance recognition of exonic regions.

Results for A. thaliana: (Comparison with known gene models)

	transcript level $(SN + SP)/2$
1. mGene (*ab initio*) ...	**73.3%**
2. ... with tiling arrays (11 tissues)	**82.1%**
3. ... with mRNA-seq (1 tissue)	**81.1%**

Transcriptome measurements certainly help, but predictions are still quite far from perfect.

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

Idea: Use transcriptome measurements to enhance recognition of exonic regions.

Results for A. thaliana: (Comparison with known gene models)

	transcript level $(SN + SP)/2$
1. mGene (*ab initio*) ...	**73.3%**
2. ... with tiling arrays (11 tissues)	**82.1%**
3. ... with mRNA-seq (1 tissue)	**81.1%**

Transcriptome measurements certainly help, but predictions are still quite far from perfect.

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

Summary and Future Work

- Genome Annotation is a huge structured output learning problem
- Proposed a two-step learning procedure separating the kernels from the structured output prediction
- Sequence classification already challenging (large!)
- String data structures make training feasible
- Gene prediction is more difficult in reality
  - Predict splice graphs/alternative transcripts
  - Regulation!? Auxiliary data!?
- New sequencing data gives
  - detailed picture on transcriptome
  - hope to be able to train more sophisticated models

Summary and Future Work

- Genome Annotation is a huge structured output learning problem
- Proposed a two-step learning procedure separating the kernels from the structured output prediction
- Sequence classification already challenging (large!)
- String data structures make training feasible
- Gene prediction is more difficult in reality
  - Predict splice graphs/alternative transcripts
  - Regulation!? Auxiliary data!?
- New sequencing data gives
  - detailed picture on transcriptome
  - hope to be able to train more sophisticated models

Summary and Future Work

- Genome Annotation is a huge structured output learning problem
- Proposed a two-step learning procedure separating the kernels from the structured output prediction
- Sequence classification already challenging (large!)
- String data structures make training feasible
- Gene prediction is more difficult in reality
  - Predict splice graphs/alternative transcripts
  - Regulation!? Auxiliary data!?
- New sequencing data gives
  - detailed picture on transcriptome
  - hope to be able to train more sophisticated models

Summary and Future Work

- Genome Annotation is a huge structured output learning problem
- Proposed a two-step learning procedure separating the kernels from the structured output prediction
- Sequence classification already challenging (large!)
- String data structures make training feasible
- Gene prediction is more difficult in reality
  - Predict splice graphs/alternative transcripts
  - Regulation!? Auxiliary data!?
- New sequencing data gives
  - detailed picture on transcriptome
  - hope to be able to train more sophisticated models

Acknowledgments

Friedrich Miescher Laboratory
of the Max Planck Society

Sequence Analysis

Sören Sonnenburg (FML/FIRST)
Gabi Schweikert (FML/MPI)
Alex Zien (FML & FIRST)
Konrad Rieck (FIRST)

Gene Finding

Gabi Schweikert (FML/MPI)
Jonas Behr (FML)
Alex Zien (FML & FIRST)
Georg Zeller (FML/MPI)

Short Reads

Fabio De Bona (FML)
Jonas Behr (FML)
Detlef Weigel (MPI)

More Information

- http://fml.mpg.de/raetsch
- http://shogun-toolbox.org
- http://mgene.org/web
- Slides with references are available online

Thank you!

Acknowledgments

Friedrich Miescher Laboratory
of the Max Planck Society

Sequence Analysis

Sören Sonnenburg (FML/FIRST)
Gabi Schweikert (FML/MPI)
Alex Zien (FML & FIRST)
Konrad Rieck (FIRST)

Gene Finding

Gabi Schweikert (FML/MPI)
Jonas Behr (FML)
Alex Zien (FML & FIRST)
Georg Zeller (FML/MPI)

Short Reads

Fabio De Bona (FML)
Jonas Behr (FML)
Detlef Weigel (MPI)

More Information

- `http://fml.mpg.de/raetsch`
- `http://shogun-toolbox.org`
- `http://mgene.org/web`
- Slides with references are available online

Thank you!

Acknowledgments

Friedrich Miescher Laboratory
of the Max Planck Society

Sequence Analysis

Sören Sonnenburg (FML/FIRST)
Gabi Schweikert (FML/MPI)
Alex Zien (FML & FIRST)
Konrad Rieck (FIRST)

Gene Finding

Gabi Schweikert (FML/MPI)
Jonas Behr (FML)
Alex Zien (FML & FIRST)
Georg Zeller (FML/MPI)

Short Reads

Fabio De Bona (FML)
Jonas Behr (FML)
Detlef Weigel (MPI)

More Information

- `http://fml.mpg.de/raetsch`
- `http://shogun-toolbox.org`
- `http://mgene.org/web`
- Slides with references are available online

Thank you!

References I

Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden Markov Support Vector Machines. In *Proc. 20th Int. Conf. Mach. Learn.*, pages 3–10, 2003.

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, May 2008.

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.

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.

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.

V. Franc and S. Sonnenburg. Optimized cutting plane algorithm for large-scale risk minimization. *Journal of Machine Learning Research*, 2009. in press.

S.S. Gross, O. Russakovsky, C.B. Do, and S. Batzoglou. Training conditional random fields for maximum labelwise accuracy. In *Proc. NIPS 2006*, 2006.

References II

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *In International Conference on Machine Learning*. 2001.

L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

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 and M.K. Warmuth. Efficient margin maximization with boosting. *Journal of Machine Learning Research*, 6:2131–2152, 2005.

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

G. Rätsch, S. Sonnenburg, J. Srinivasan, H. Witte, K.R. Müller, R. Sommer, and B. Schölkopf. Improving the caenorhabditis elegans genome annotation using machine learning. *PLoS Comput Biol*, 3(2):e20, 2007.

G. Schweikert, G. Zeller, A. Zien, J. Behr, C.S. Ong, P. Philips, A. Bohlen, R. Bohnert, F. De Bona, S. Sonnenburg, and G. Rätsch. mGene: Accurate computational gene finding with application to nematode genomes. under revision, March 2009a.

References III

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, 2009b. 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*, 2009c. 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 Kernel Machines*, chapter Large Scale Learning with String Kernels. MIT Press, 2007a.

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.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Large-margin methods for structured output spaces. *Journal for Machine Learning Research*, 6:1453–1484, September 2005.

References IV

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, 2008a. ISSN 1088-9051 (Print). doi: 10.1101/gr.070169.107.

G. Zeller, S.R. Henz, S. Laubinger, D. Weigel, and G Rätsch. Transcript normalization and segmentation of tiling array data. In *Proceedings Pac. Symp. on Biocomputing*, pages 527–538, 2008b.

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.

A. Zien, C.S. Ong, and G. Rätsch. Towards the inference of graphs on ordered vertices. Research Note 150, Max Planck Institute for biological Cybernetics, Tübingen, Germany, 2006. URL `http://www.kyb.mpg.de/publication.html?publ=4133`.