New Methods for Splice Site Recognition

Sören Sonnenburg[†], Gunnar Rätsch[♮], Arun Jagota[♯] and Klaus-Robert Müller[‡]

† Fraunhofer FIRST, Kekuléstr. 7, 12489 Berlin, Germany
♮ The Australian National University, Canberra, ACT 0200, Australia
♯ University of California at Santa Cruz, CA 95064, USA
‡ University of Potsdam, August-Bebel-Str. 89, 14482 Potsdam, Germany

{Soeren.Sonnenburg, Klaus-Robert.Mueller}@first.fraunhofer.de,
Gunnar.Raetsch@anu.edu.au, jagota@cse.ucsc.edu

Roadmap: Classification of Splice Sites

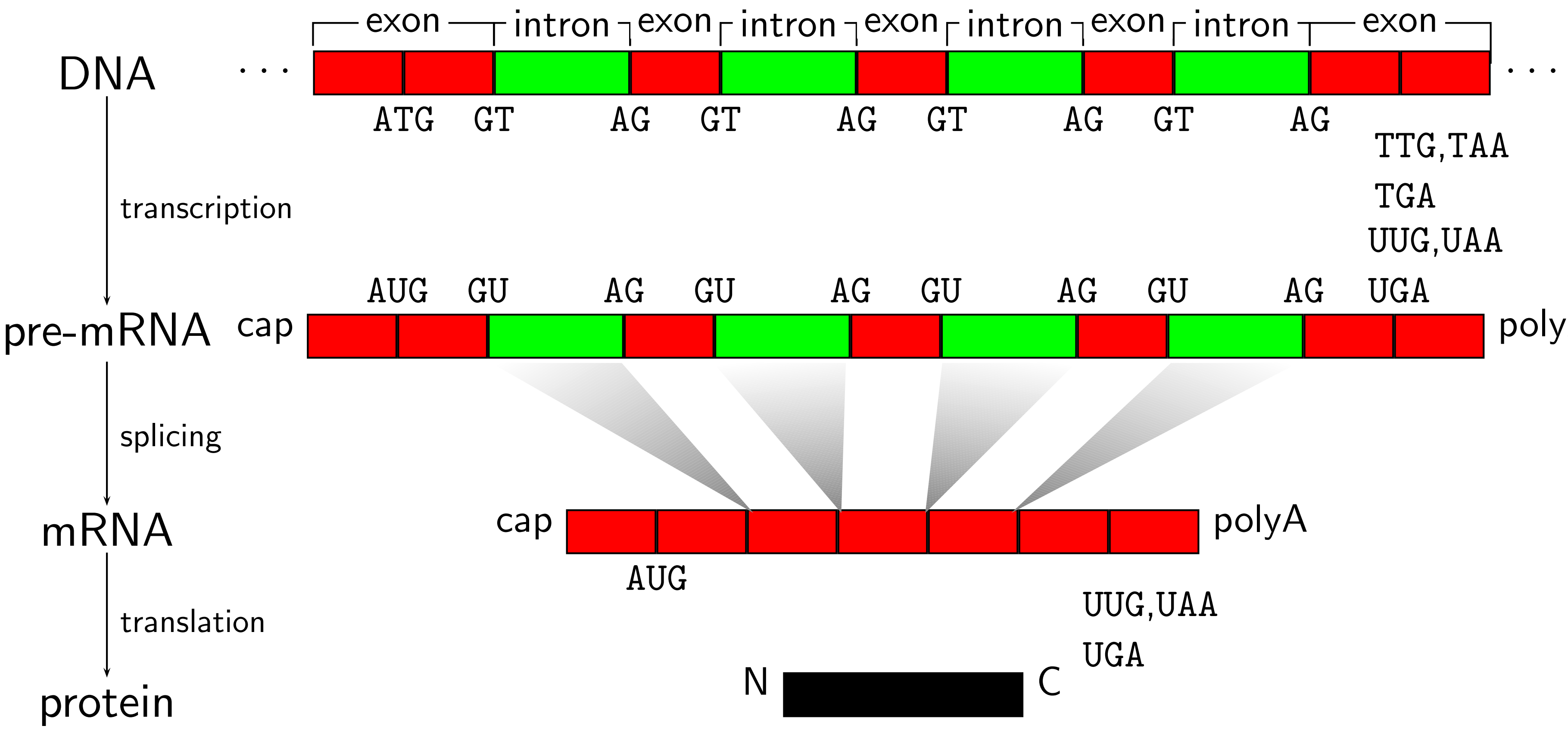
...GTTGACGATCGAGTACGCACAAGCTCAGGAGTCCAGCGGTGAAGAGAGGTTAAGCTCGTCGCTGCT...

- Biological Introduction to Splice Sites
- Prior Knowledge for Splicing and ML
  1. engineering
  2. generative
- Benchmarking on the IPData dataset (human)
- Experiments on the *C. elegans* genome
- Conclusion

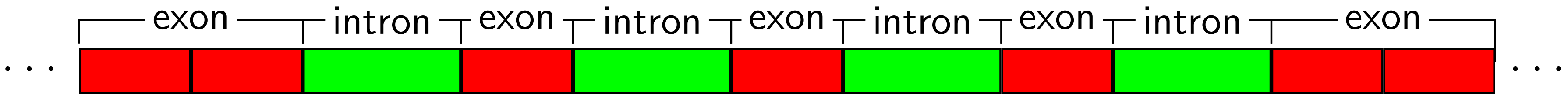
Aim: Improve the splicing module in a gene finder

Biological Background

Splice sites are locations on DNA at boundaries of
- exons (which code for proteins)
- introns (which do not)



Facts about Splice Sites



- Exons are short ($100 - 200$ bp); Introns can be very long (> 1 kbp)
- The splicing process takes place in the cell's *nucleus*.
- The apparatus for splicing (“**Spliceosome**”) is not tissue specific
- Splicing mechanisms are very similar for all higher organism
- Experiments show that any 5' site could be connected to any 3' site

The splicing mechanism uses *local* information

Why is Splice Site Detection important?

- allows to *accurately* predict mRNA and thus proteins from DNA

  ⇒ important step in analyzing the genome

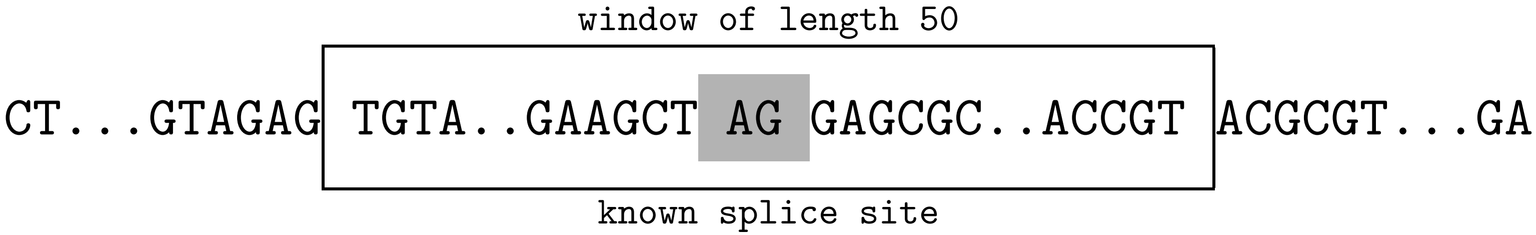
- splice sites can be detected with high accuracy

  ⇒ important and accurate ‘marker‘ to find locations of genes

Conventional: alignment to data base entries

Aim: Improve the Splicing module with ML

Two-Class Classification Problem



window of length 50

CT...GTAGAG TGTA..GAAGCT AG GAGCGC..ACCGT ACGCGT...GA

known splice site

- only considered canonical splice sites (consensus AG,GT, 98%)
- true splice sites: fixed window around splice site
- decoys: created by sliding the window ±25 bases

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

Download at http://mlg.anu.edu.au/~raetsch/splice

Support Vector Machines

- sequences $\boldsymbol{x}_i \in \mathbb{X}$ $(i = 1, \ldots, \ell)$ with respective labels y_i
- SVM classifier (essentially perceptron in kernel feature space):

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

- find parameters $\boldsymbol{\alpha}$ by solving quadratic optimization problem:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \, \mathrm{k}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

subject to $\alpha_i \in [0, C]$, $i = 1, \ldots, \ell$, $\sum_{i=1}^{\ell} \alpha_i y_i = 0$.

Solution has no local minima

Engineering Kernels I

Polynomial Kernel of degree d:

$$\mathrm{k}_{\mathrm{POLY}}(\boldsymbol{x}, \boldsymbol{x}') = \left(\sum_{p=1}^{l} \mathsf{I}_p(\boldsymbol{x}, \boldsymbol{x}')\right)^d$$

$\Rightarrow$ Computes all d-th order monomials:

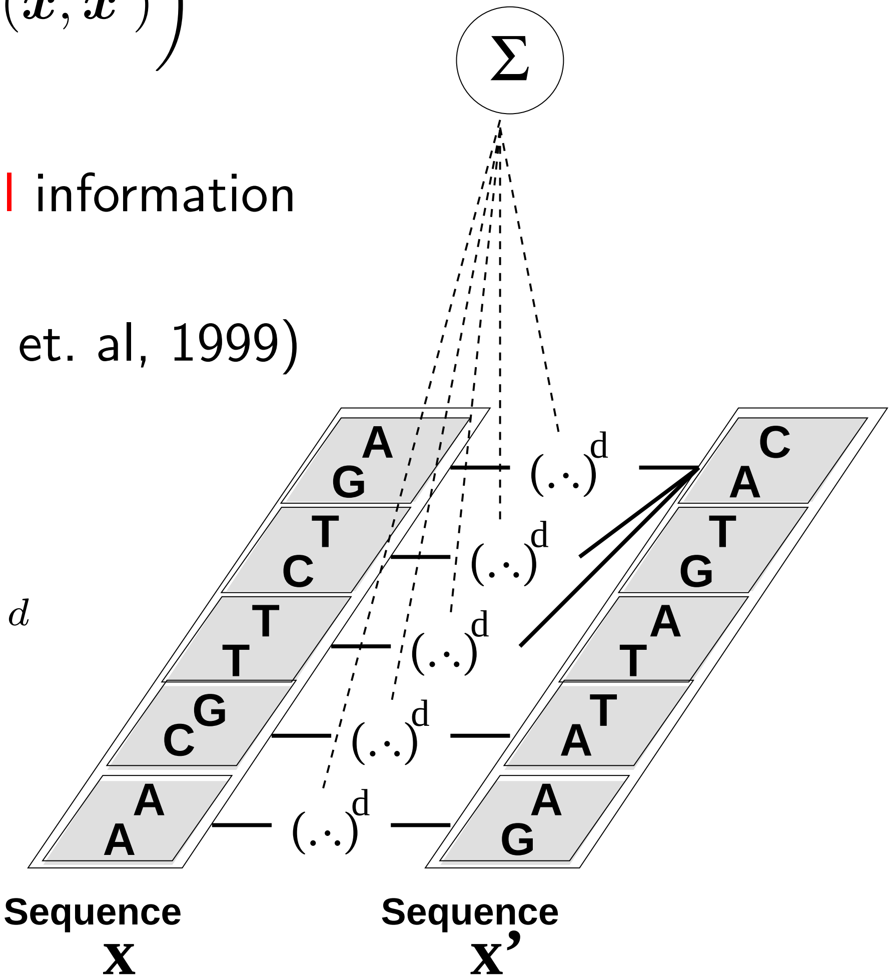
uses global information

Locality Improved Kernel (Zien et. al, 1999)

$$\mathrm{k}_{\mathrm{LI}}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{p=1}^{N} \mathsf{win}_p(\boldsymbol{x}, \boldsymbol{x}')$$

$$\mathsf{win}_p(\boldsymbol{x}, \boldsymbol{x}') = \left(\sum_{j=-l}^{+l} p_j \mathsf{I}_{p+j}(\boldsymbol{x}, \boldsymbol{x}')\right)^d$$

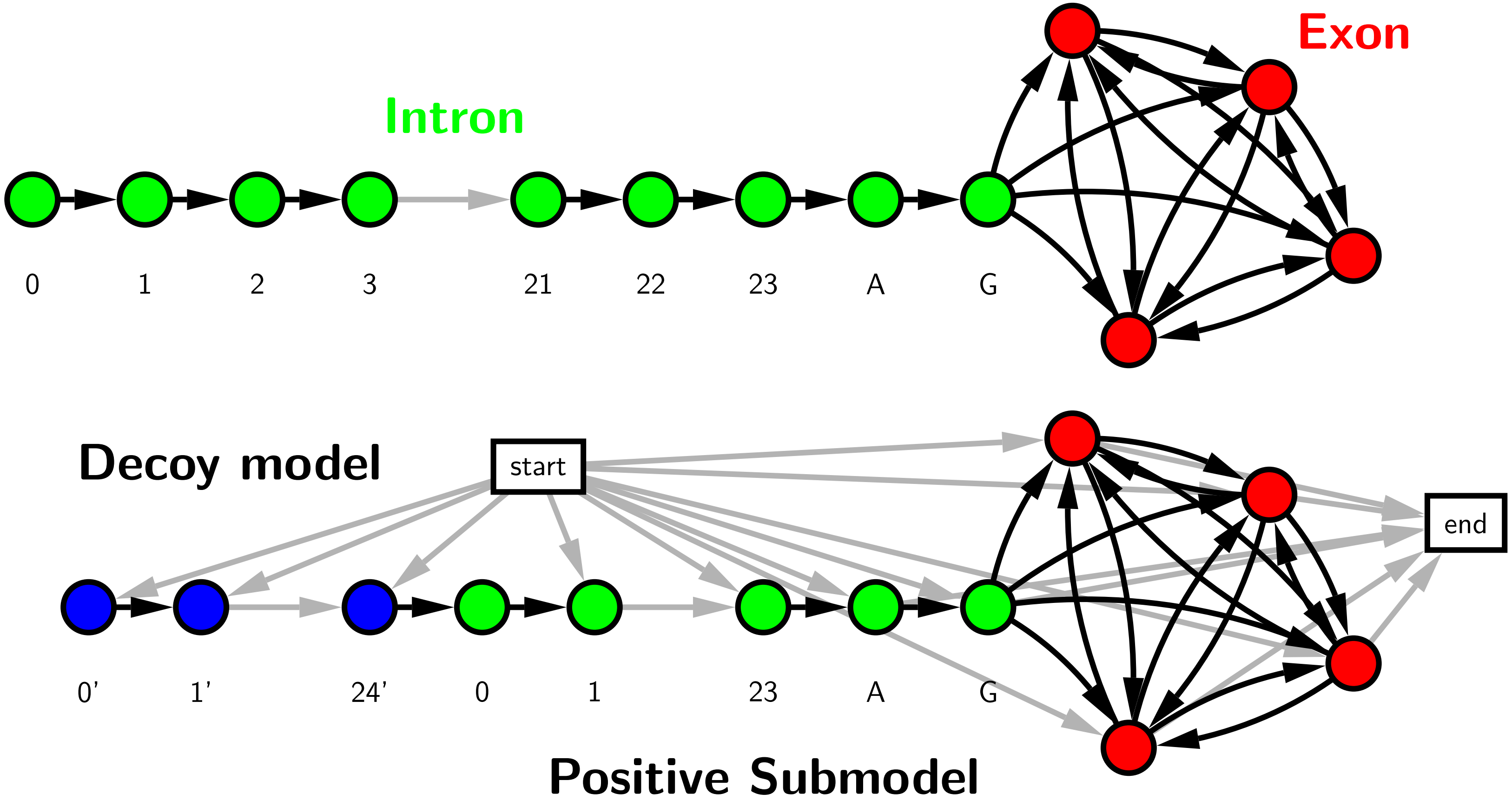
$$\mathsf{I}_i(x, x') = \begin{cases} 1, & x_i = x'_i \\ 0, & \text{otherwise} \end{cases}$$



Idea: Spliceosome uses only local information → we need local classifiers

Generative Models

Use generative model, e.g. design a HMM



(top) positive acceptor model, (bottom) negative acceptor model

Engineering Kernels II

- Kernels from generative models
  - compare objects using a generative model $\mathrm{Pr}(\boldsymbol{x}|\Theta)$
  - exploit probabilistic model for discriminative training

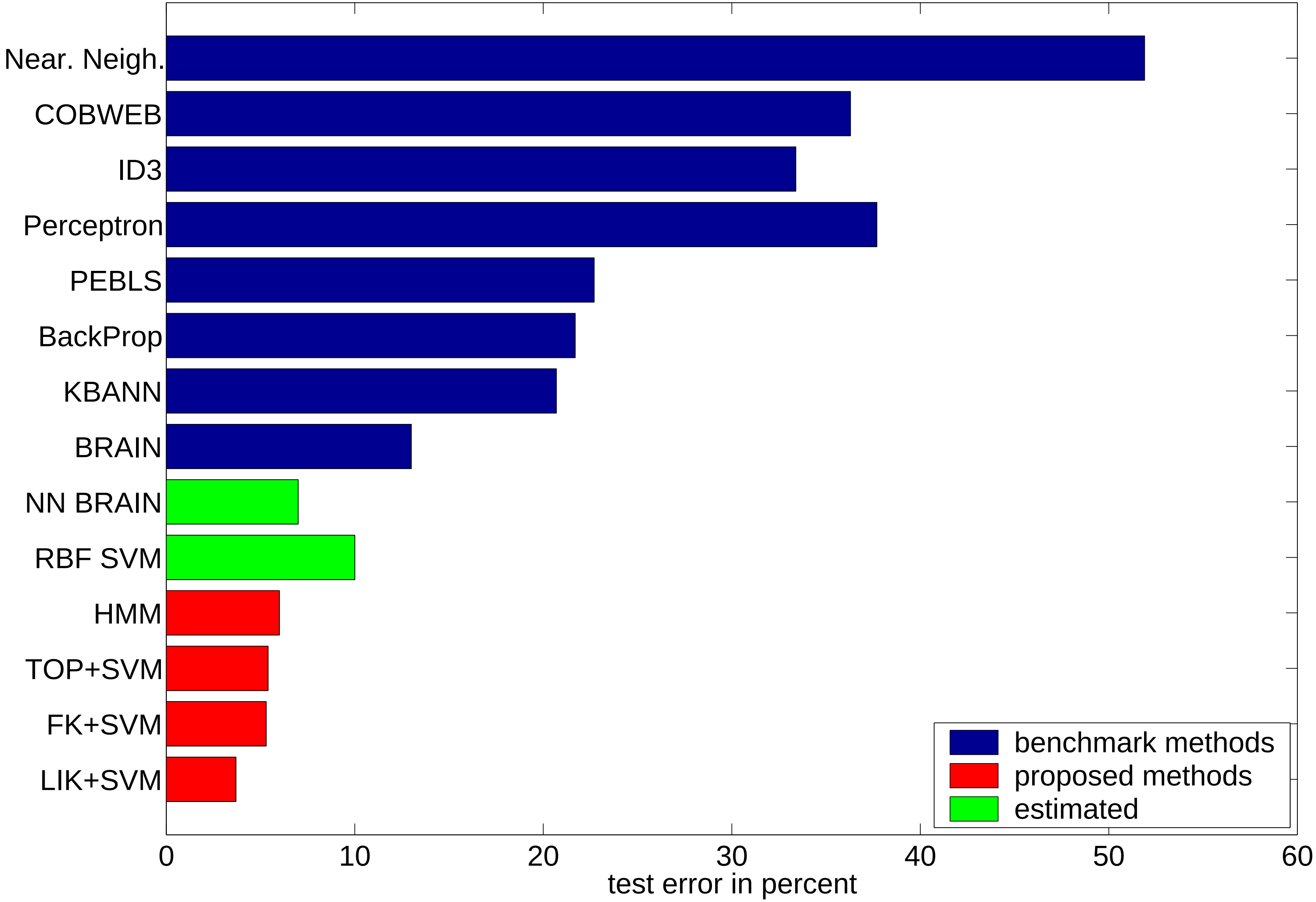
- Fisher Kernel (Jaakkola and Haussler, 1998)

$$\begin{aligned} \mathrm{k}_{\mathrm{FK}}(\boldsymbol{x},\boldsymbol{x}') &= \boldsymbol{s}(\boldsymbol{x},\hat{\boldsymbol{\theta}})^\top Z^{-1}(\hat{\boldsymbol{\theta}})\boldsymbol{s}(\boldsymbol{x}',\hat{\boldsymbol{\theta}}) \\ \boldsymbol{s}(\boldsymbol{x},\hat{\boldsymbol{\theta}}) &= \nabla_{\boldsymbol{\theta}} \log \mathrm{Pr}(\boldsymbol{x}|\hat{\boldsymbol{\theta}}) && \text{Fisher score vector} \\ Z(\hat{\boldsymbol{\theta}}) &= \mathrm{E}_{\boldsymbol{x}}\left[\boldsymbol{s}(\boldsymbol{x},\hat{\boldsymbol{\theta}})\boldsymbol{s}(\boldsymbol{x},\hat{\boldsymbol{\theta}})^\top \middle| \hat{\boldsymbol{\theta}}\right] && \text{Fisher information matrix} \end{aligned}$$

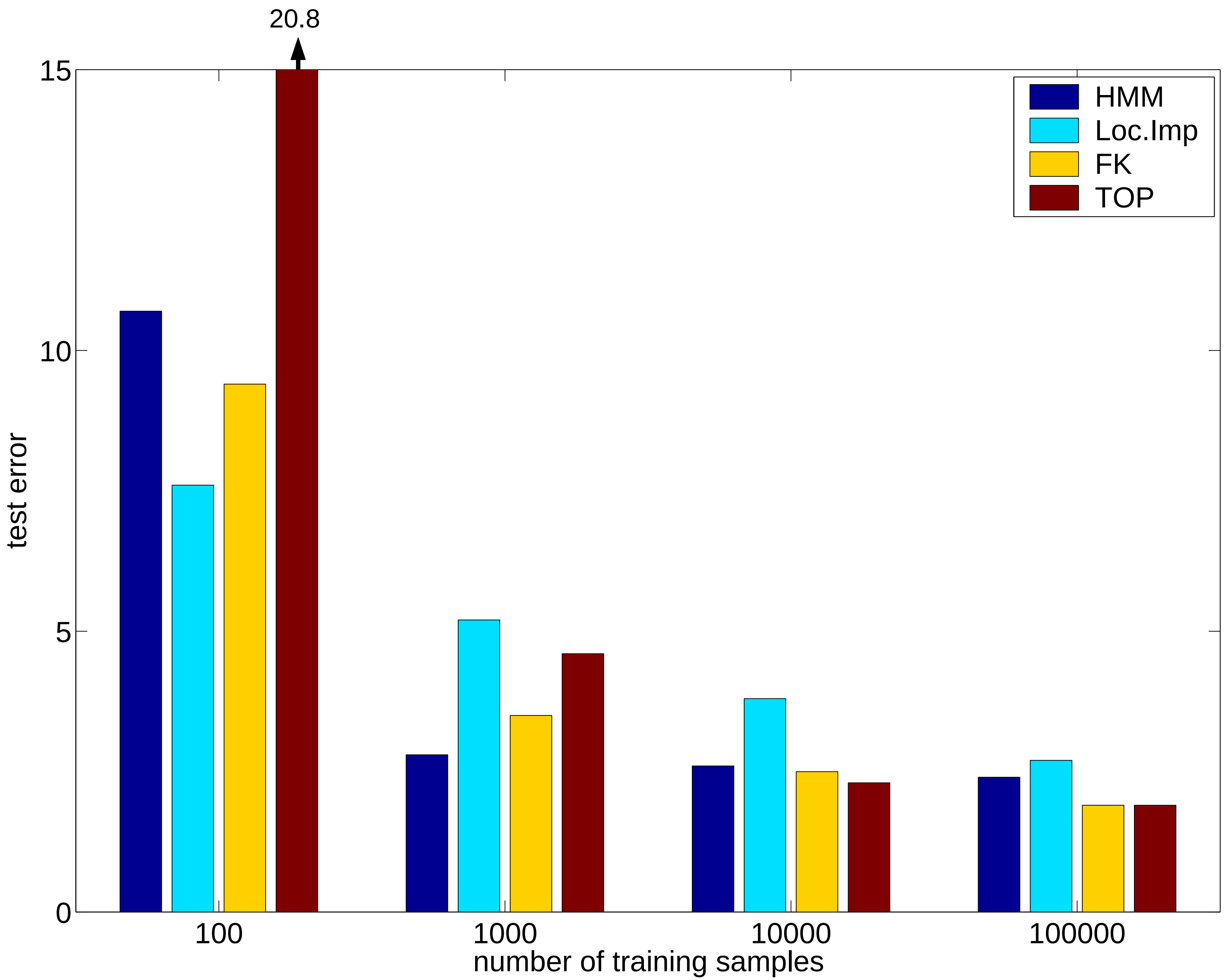
- TOP Kernel (Tsuda et. al, 2002)

$$\begin{aligned} \mathrm{k}_{\mathrm{TOP}}(\boldsymbol{x},\boldsymbol{x}') &= \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(x)^\top \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(x') \\ \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(x) &= (v(\boldsymbol{x},\hat{\boldsymbol{\theta}}), \nabla_\theta\, v(\boldsymbol{x},\hat{\boldsymbol{\theta}}))^\top \\ v(\boldsymbol{x},\hat{\boldsymbol{\theta}}) &= \log(\mathrm{Pr}(y=+1|\boldsymbol{x},\hat{\boldsymbol{\theta}})) - \log(\mathrm{Pr}(y=-1|\boldsymbol{x},\hat{\boldsymbol{\theta}})) \end{aligned}$$

Benchmark Results on IPData (human genome)



Results on *C. elegans* acceptor sites



Conclusion

- 2 ways to engineer kernels (locality improved, from generative model)
- application to splice site recognition (record performance)
- benchmark results (human) and result on *C. elegans* genome
- computing time 30 CPU years (APAC Super Computer)
- Philosophical issues:

  1. explicit use of biological prior knowledge vs. use of generative model?
  2. discriminative training vs. generative models

For more information, datasets, related papers visit:

`http://mlg.anu.edu.au/~raetsch/splice/`