Large Scale Learning with String Kernels

Sören Sonnenburg
Fraunhofer FIRST.IDA, Berlin

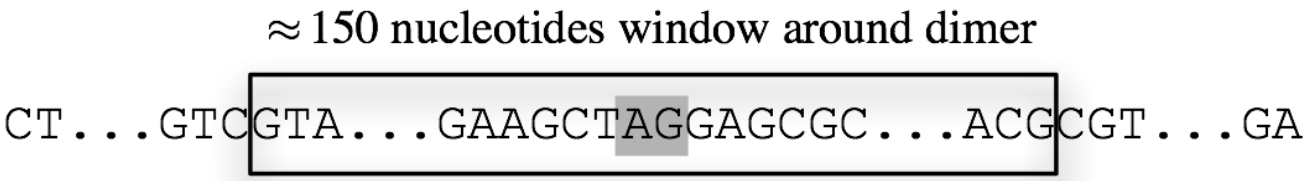
joint work with
Vojtech Franc, Alexander Zien, Gunnar Rätsch

Fraunhofer Institut Rechnerarchitektur und Softwaretechnik

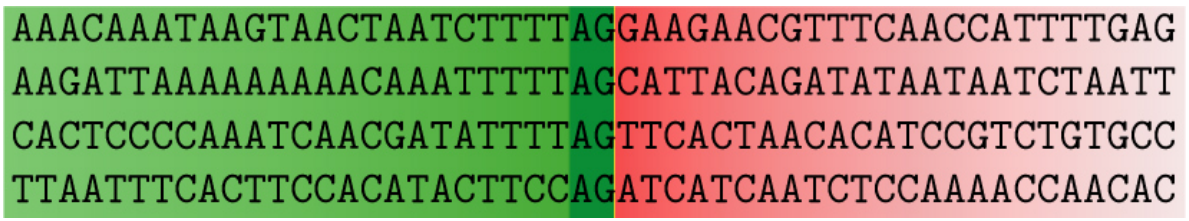
Friedrich Miescher Laboratory
of the Max Planck Society

The Motivating Application - Splice Site recognition

Discriminate true signal positions against all other positions



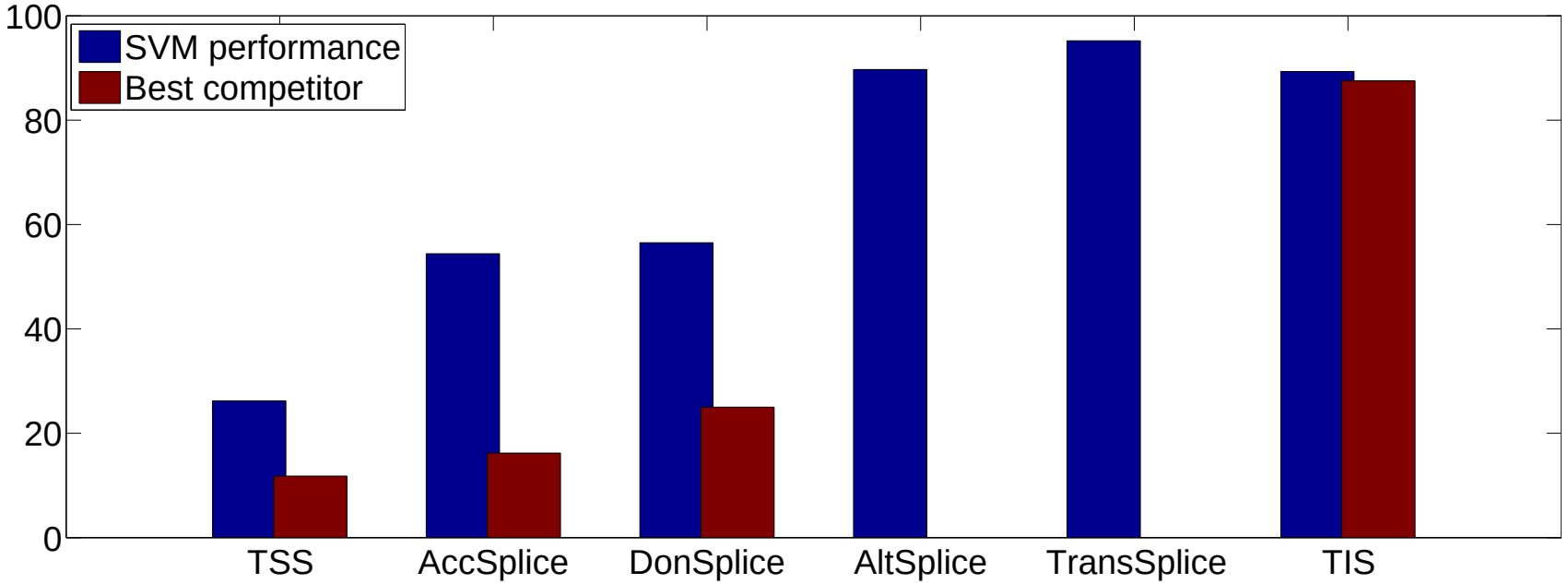
- True sites: fixed window around a true splice site
- Decoy sites: all other consensus sites



- Sequences are compared via String-Kernels
  - For each position a Weighted Degree Kernel compares all k-mers up to maximal length K

SVM ≈ 3 times more accurate than IMCs
(54.4% vs. 16.2% auPRC)

Beauty in Generality

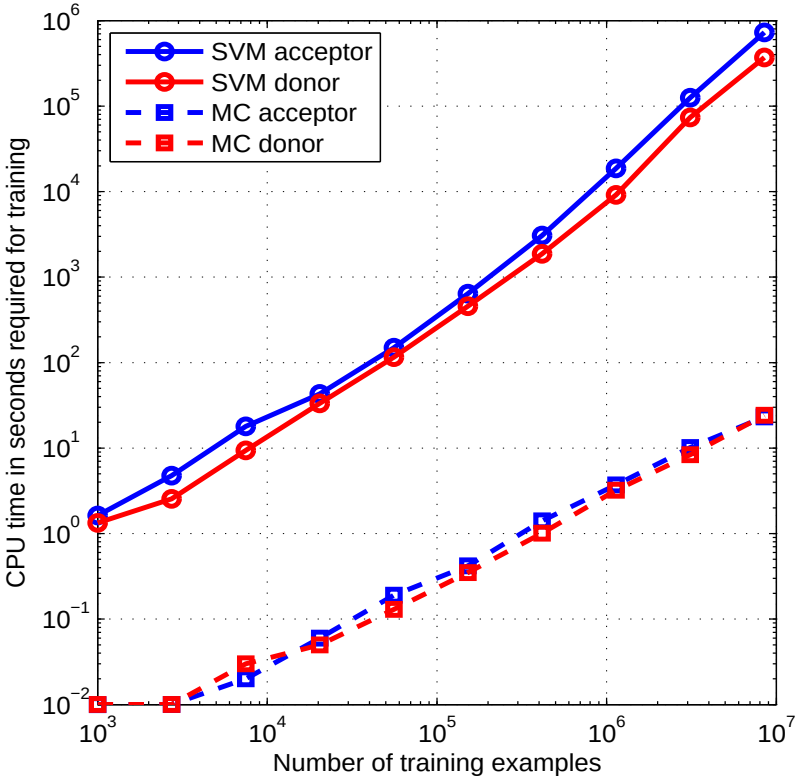


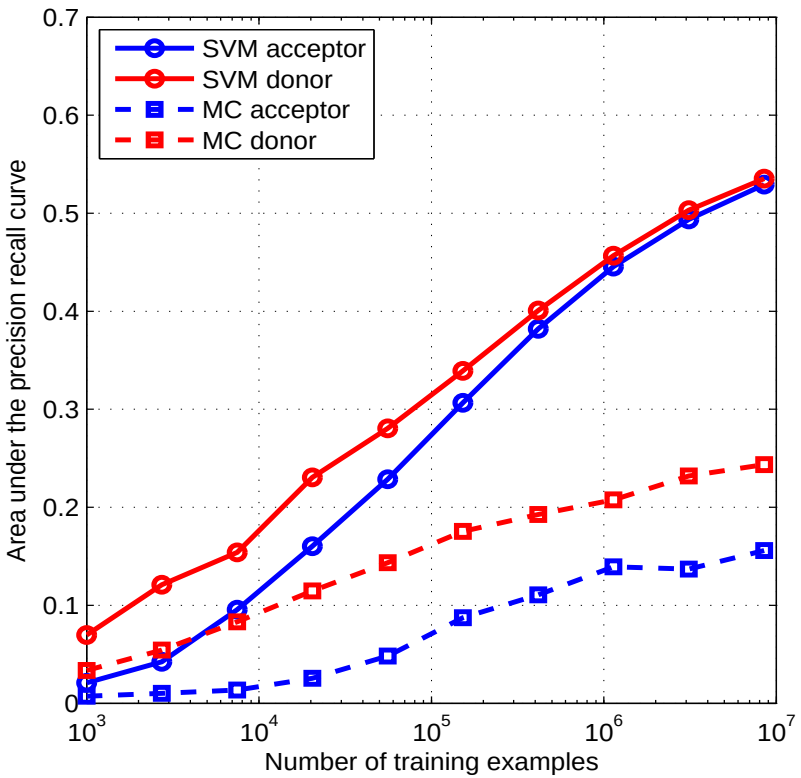
- Transcription Start (Sonnenburg et al., Eponine Down et al.)
- Acceptor Splice Site (Philips et al.)
- Donor Splice Site (Philips et al.)
- Alternative Splicing (Rätsch et al., -)
- Transsplicing (Schweikert et al., -)
- Translation Initiation (Sonnenburg et al., Saeys et al.)

Drawback: SVM Training Time too large

Task

- Human splice sites: $5 \cdot 10^7$ strings of length ≈ 200
- Note: Raw data is already 9GB in size

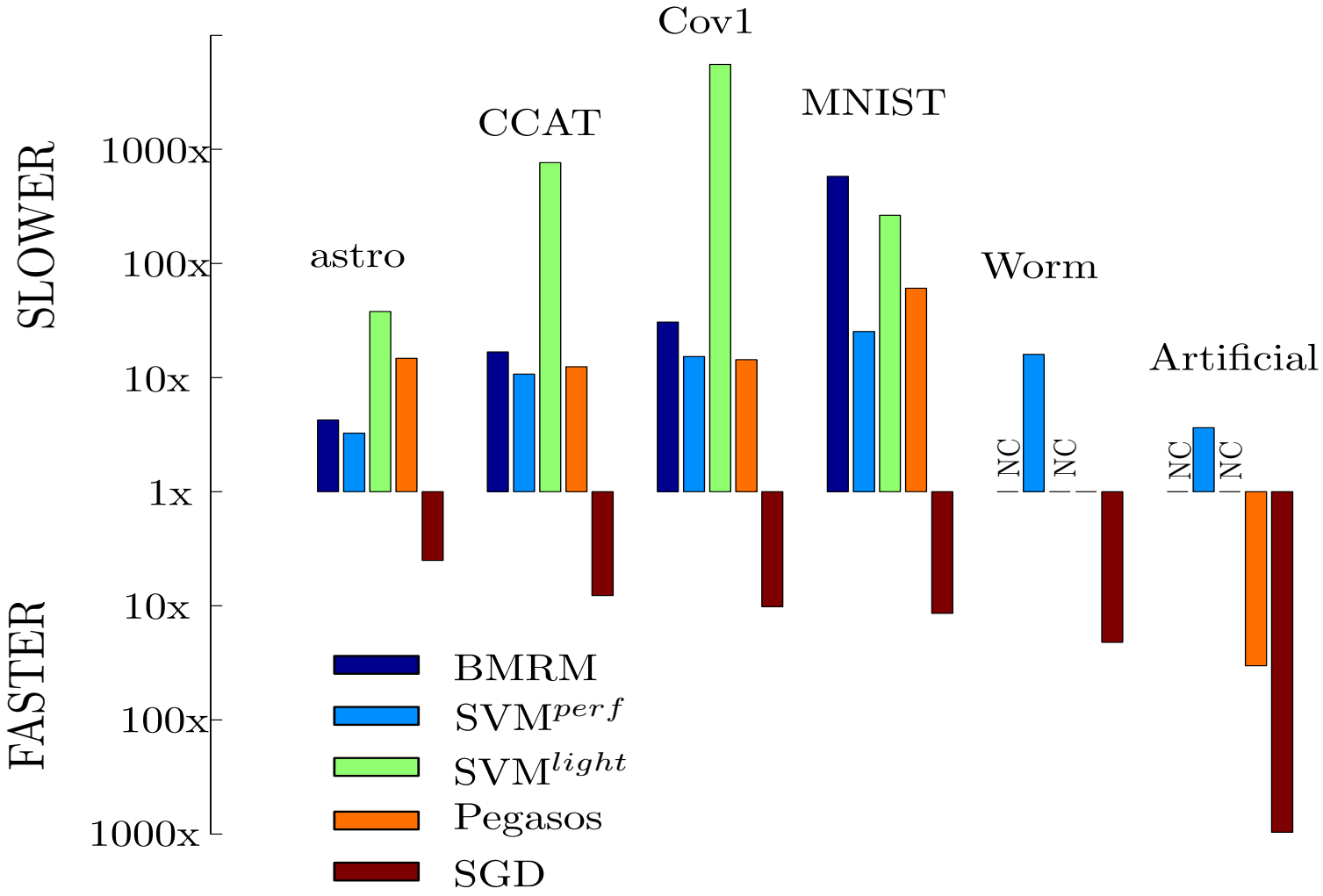




Train string-kernel SVM on all available data

Idea - Use recent advances in learning *linear* SVMs

Optimized Cutting Plane SVM (OCAS, Franc et al.)



Effort $\mathcal{O}(N)$ instead of $\mathcal{O}(N^2)$ to $\mathcal{O}(N^3)$

String Kernel SVM Scoring Function

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

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

Training Linear SVMs in String Kernel Feature Space

Mapping

- A-priory compute $\Phi(\mathbf{x}_i)$ for all $i = 1, \ldots, N$
- Train linear SVM classifier using, e.g., OCAS

Limitations

1. Only small order possible
   ($K = 8 \Rightarrow$ 87,380 dimensions, per position)
2. Explicit (sparse) representation too costly

Simple Trick

- Only $\mathbf{w} \leftarrow \mathbf{w} + \alpha\Phi(\mathbf{x})$ and $\mathbf{w} \cdot \Phi(\mathbf{x})$ required

$\Rightarrow$ Compute $\Phi(\mathbf{x})$ on-the-fly and parallelize!

Preliminary Results

Human splice dataset

1. Linear SVM in primal (OCAS, 4 CPUs) (**proposed method**)
   - $5 \cdot 10^7$ strings of length 140
   - WD kernel of order 6 (and 8)
   - Dimensionality of $\mathbf{w}$ is $12 \cdot 10^6$ (requires 80M)

N	Time	auPRC
10^6	90 sec	34.35%
10^7	70 min	47.47%
$5 \cdot 10^7$	8 hrs	53.92%

2. WD Kernel order > 20 (SVM^{light} + `linadd`, ≥ 4 CPUs)

N	Time	auPRC
10^6	≈ 3 hrs	≈ 45.00
$8 \cdot 10^6$	> 8 days	54.12

Discussion

Conclusions

- Simple speedup trick for string kernels
- Shared memory parallelization, able to train on **50 million** human splice sites with **12 million** dimensions within a day

Discussion

- Limited to low $K-$mer length
- Is large K really necessary biology-wise?
- How to improve feature space?

Implemented in SHOGUN `http://www.shogun-toolbox.org`