Positional Oligomer Importance Matrices

(Feature Extraction & Interpretable SVMs)

Sören Sonnenburg

Fraunhofer FIRST.IDA, Berlin

joint work with

Alexander Zien, Petra Philips and Gunnar Rätsch

FIRST

Fraunhofer Institut Rechnerarchitektur und Softwaretechnik

Outline

The Motivating Application - Sequence Classification

```
AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC
TTGTTTTAATATTCAATTTTTTACAGTAAGTTGCCAATTCAATGTTCCAC
TACCTAATTATGAAATTAAAATTCAGTGTGCTGATGGAAACGGAGAAGTC
```

SVM+String kernel(s) state of the art in detecting

- Gene Start/End
- Splice Sites
- Trans-splicing, Alternative Splicing etc. etc.

SVM sensitivity ≈ 2 times larger at same specificity

Drawback: We loose interpretability of the result!

Why are SVMs hard to interpret?

Problem: Learned α weighting of training points
But: One is interested in discriminating features

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

Idea: Use SVMs w vector to interpret features

SVMs are interpretable - depending on feature space

Condition:

Feature space enumerable/meaningful/w storable

- linear SVMs
- most of string kernels ($k-$mer based)
  - spectrum kernel
  - WD kernels
  - ...

Problems:

- Feature space may be very high dimensional
- Features not independent

Idea:

- Compute expected SVM output given a certain feature

The Weighted Degree Kernel

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

```
x         AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG
#1-mers   .|.|.|||.|..||.|.|..|||.||...|....|...|||......|..
#2-mers   .....||.....|.......||..|.............||.........
#3-mers   .....|..............|.................|.........
y         TACCTAATTATGAAATTAAATTTCAGTGTGCTGATGGAAACGGAGAAGTC
```

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

The Scoring System - Definition

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	$\cdots$

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

The Scoring System - Examples

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

Examples:

- WD-kernel
- WD-kernel with shifts
- Spectrum kernel
- Oligo Kernel

Not limited to SVMs:

- markov chains (higher order/inhomogeneous/mixed order)

POIMs

Idea:

- Compute expected score $C(\mathbf{z}, j)$ given that $k-$mer $\mathbf{z}$ appears at position j in the sequence for small k
- normalized with expected score over all sequences

$$C(\mathbf{z}, j) \; := \; \mathbb{E}\left[s(\mathbf{x}) \mid \mathbf{x}[j] = \mathbf{z} \right] - \mathbb{E}\left[s(\mathbf{x}) \right]. \qquad (1)$$

Problem:

- Choosing a background distribution for $\mathbf{x}$ (uniform, 0−th order MC)
- Naive approach already for short sequences and small alphabets infeasible
- $\mathbf{w}$ may be stored in some sparse data structure (like a tree/forest)

Needs efficient algorithm for computation

Observations

$$
\begin{aligned}
C(\mathbf{z},j) &:= \mathbb{E}\left[\, s(\mathbf{x}) \mid \mathbf{x}[j] = \mathbf{z}\,\right] - \mathbb{E}\left[\, s(\mathbf{x})\,\right]. \\
&= \sum_{(\mathbf{y},i)\in\mathcal{I}} w(\mathbf{y},i)\left[Pr\left(\mathbf{x}[i] = \mathbf{y} \mid \mathbf{x}[j] = \mathbf{z}\right) - Pr\left(\mathbf{x}[i] = \mathbf{y}\right)\right] \\
&= u(\mathbf{z},j) + \sum_{\mathbf{y}\in\Sigma^{|\mathbf{z}|}} u(\mathbf{z},j)
\end{aligned}
$$

- number of $k-$mers grows only linear with data
- all features which are independent of $(\mathbf{z},j)$ vanish
- computation can be split in computing contributions from 4 cases:

$$
\begin{aligned}
u(\mathbf{z},j) &:= \sum_{(\mathbf{y},i)\in\mathcal{I}(\mathbf{z},j)} Pr\left(\mathbf{x}[i] = \mathbf{y} \mid \mathbf{x}[j] = \mathbf{z}\right) w(\mathbf{y},i) \\
&= u^{\vee}(\mathbf{z},j) + u^{\wedge}(\mathbf{z},j) + u^{<}(\mathbf{z},j) + u^{>}(\mathbf{z},j) - w(\mathbf{z},j) \;,
\end{aligned}
$$

TACGT	...AATACGTAC...	...AATACGT	TACGTAC...
AATACGTAC	AATACGTAC	AATACGTAC	AATACGTAC

For `AATACGTAC`: substring, superstring, left and right partial overlap

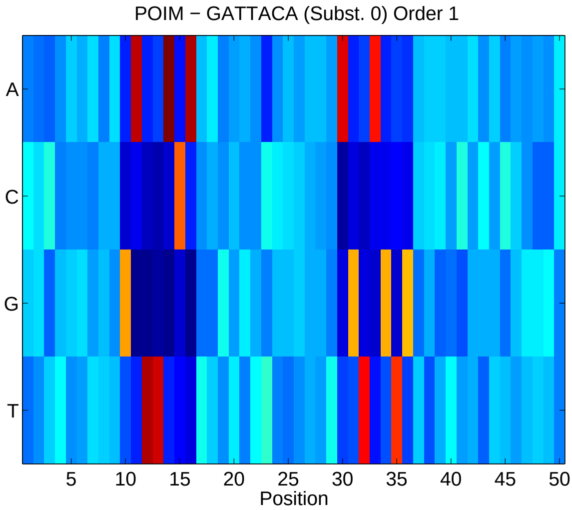
Efficient Computation

$$
\begin{aligned}
u^{\vee}(\mathbf{z},j) &= w(\mathbf{z},j) + u^{\vee}(\tau\mathbf{z}',j) + u^{\vee}(\mathbf{z}'\tau',j+1) - u^{\vee}(\mathbf{z}',j+1) \\
u^{\wedge}(\mathbf{z},j) &= w(\mathbf{z},j) - \sum_{(\sigma,\sigma')\in\Sigma^2} Pr\left(\mathbf{x}[j+|\mathbf{z}|]=\sigma'\right) Pr\left(\mathbf{x}[j-1]=\sigma\right) u^{\wedge}(\sigma\mathbf{z}\sigma',j-1) \\
&+ \sum_{\sigma\in\Sigma} Pr\left(\mathbf{x}[j-1]=\sigma\right) u^{\wedge}(\sigma\mathbf{z},j-1) + \sum_{\sigma'\in\Sigma} Pr\left(\mathbf{x}[j+|\mathbf{z}|]=\sigma'\right) u^{\wedge}(\mathbf{z}\sigma',j) \\
u^{<}(\mathbf{z},j) &= \sum_{\sigma\in\Sigma} Pr\left(\mathbf{x}[j-1]=\sigma\right) \sum_{k=1}^{|\mathbf{z}|-1} L(\sigma\mathbf{z}[1]^k,j-1) \\
u^{>}(\mathbf{z},j) &= \sum_{\sigma\in\Sigma} Pr\left(\mathbf{x}[j+|\mathbf{z}|]=\sigma\right) \sum_{k=1}^{|\mathbf{z}|-1} R(\mathbf{z}[|\mathbf{z}|-k+1]^k\sigma,j+|\mathbf{z}|-k) \ ,
\end{aligned}
$$

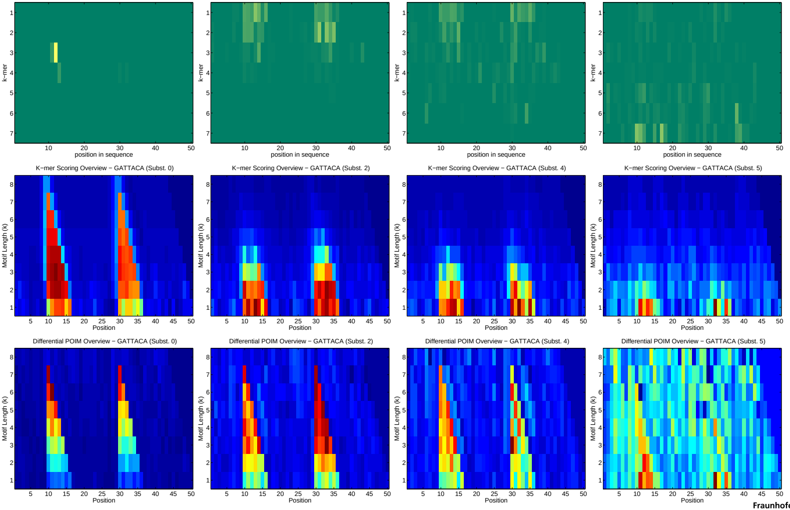
where

$$
\begin{aligned}
L(\mathbf{t},j) &:= \sum_{(\mathbf{y},i)\in\mathcal{L}(\mathbf{t},j)} Pr\left(\mathbf{x}[i]=\mathbf{y} \mid \mathbf{x}[j]=\mathbf{t}\right) w(\mathbf{y},i) \\
R(\mathbf{t},j) &:= \sum_{(\mathbf{y},i)\in\mathcal{R}(\mathbf{t},j)} Pr\left(\mathbf{x}[i]=\mathbf{y} \mid \mathbf{x}[j]=\mathbf{t}\right) w(\mathbf{y},i) \ .
\end{aligned}
$$

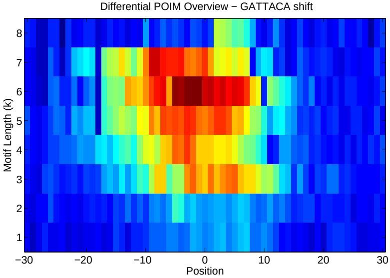
GATTACA and AGTAGTG at fixed positions 10 and 30

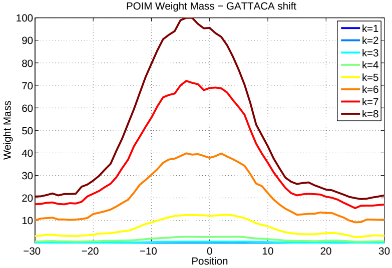


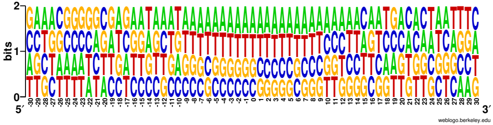
GATTACA and AGTAGTG at fixed positions 10 and 30



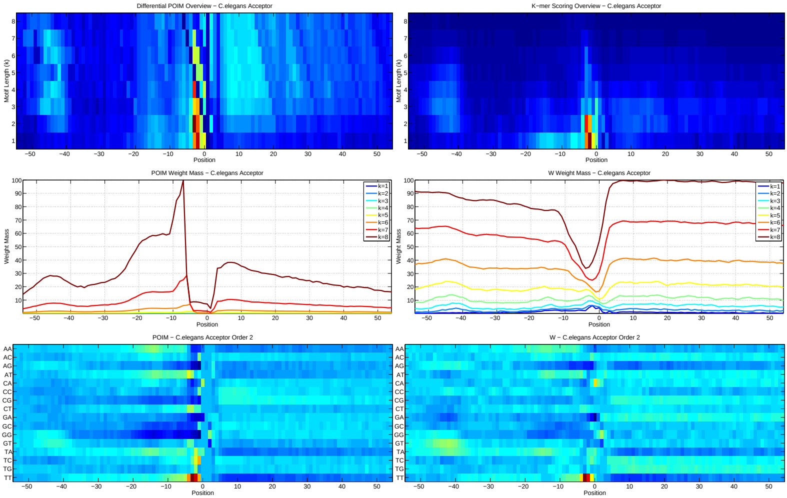
GATTACA at variable positions



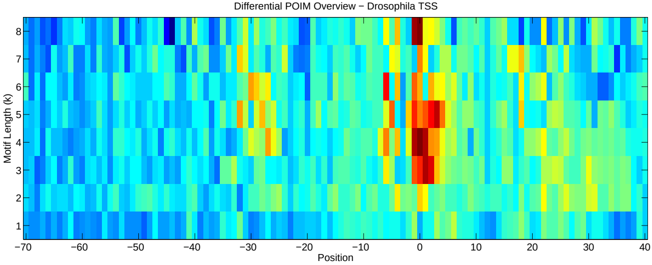


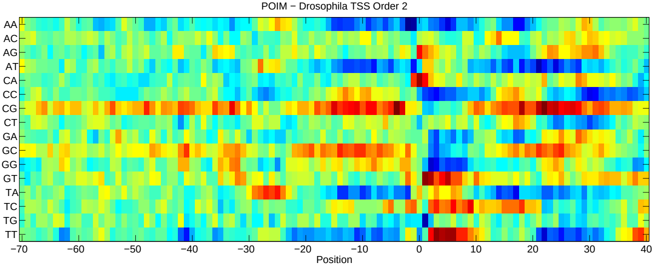


C.elegans Acceptor Splice Site Recognition



Drosophila Transcription Starts





Open Problems

- **Motif detection**
  - `GCGCG` vs `CGCGC` - how to merge?
  - strong motifs dominate weaker ones - how to cluster?
- **Correction necessary for repetitive motifs?**
  - **z** scores high **zz** will score higher
- **Consensus sequence** $\mathbf{x}^* := \operatorname{argmax}_{\mathbf{x}} s(\mathbf{x})$ - Is it meaningful?

Conclusions

Positional Oligomer Importance Matrices

- developed a method which systematically computes the importances of positional motifs for the expected decision score
  - useful to rank motifs and for visualization
  - applicable for a large class of popular scores (SVM+spec/WD/oligo kernel; markov chain)
  - efficiently implemented for spectrum and WD kernels in `http://www.shogun-toolbox.org`
- nice results on toy and real world data