Opening the Black Box: Revealing Interpretable Sequence Motifs in Kernel-Based Learning Algorithms

Marina M.-C. Vidovic[1](✉), Nico Görnitz[1], Klaus-Robert Müller[1,2](✉),
Gunnar Rätsch[3](✉), and Marius Kloft[4](✉)

[1] Berlin Institute of Technology, 10587 Berlin, Germany
marina.vidovic@ml.tu-berlin.de,
{nico.goernitz,klaus-robert.mueller}@tu-berlin.de

[2] Department of Brain and Cognitive Engineering, Korea University, Anam-dong,
Seongbuk-gu, Seoul 136-713, Republic of Korea

[3] Memorial Sloan-Kettering Cancer Center, New York, NY 10065, USA
raetsch@mskcc.org

[4] Humboldt University of Berlin, 10099 Berlin, Germany
kloft@hu-berlin.de

Abstract. This work is in the context of kernel-based learning algorithms for sequence data. We present a probabilistic approach to automatically extract, from the output of such string-kernel-based learning algorithms, the subsequences—or *motifs*—truly underlying the machine's predictions. The proposed framework views motifs as free parameters in a probabilistic model, which is solved through a global optimization approach. In contrast to prevalent approaches, the proposed method can discover even difficult, long motifs, and could be combined with any kernel-based learning algorithm that is based on an adequate sequence kernel. We show that, by using a discriminate kernel machine such as a support vector machine, the approach can reveal discriminative motifs underlying the kernel predictor. We demonstrate the efficacy of our approach through a series of experiments on synthetic and real data, including problems from handwritten digit recognition and a large-scale *human* splice site data set from the domain of computational biology.

1 Introduction

In the view of the rapidly increasing amount of data collected in science and technology, effective automation of decisions is necessary. To this end, kernel-based methods [13,17,19,26,31,32] such as support vector machines (SVM) [5,7] have found diverse applications due to their distinct merits such as the descent computational complexity, high usability, and the solid mathematical foundation [24]. Kernel-based learning allows us to obtain more complex non-linear learning machines from simple linear ones in a canonical way, since the learning and data representation processes are decoupled in a modular fashion. Yet, after more than a decade of research, kernel methods are widely considered as black boxes, and it remains an unsolved problem to make their decisions

A. Appice et al. (Eds.): ECML PKDD 2015, Part II, LNAI 9285, pp. 137–153, 2015.
DOI: 10.1007/978-3-319-23525-7_9

accessible or interpretable to domain experts. This is especially pressing in natural and life sciences, where not maximum prediction accuracy but unveiling the underlying natural principles is the foremost aim.

In several important application fields, the data exhibits an inherent sequence structure. This includes DNA sequences in genomics, text data in natural language processing, and speech data in speech recognition. A state-of-the-art approach to learn from such sequence data consists in the weighted-degree (WD) kernel [4,27,28,31] in combination with a kernel-based learning machine such as an SVM. Given two discrete sequences $x = (x_1, \ldots, x_L),\ x' = (x'_1, \ldots, x'_L) \in \mathcal{A}^L$ of length L over the alphabet $\mathcal{A}$ with $|\mathcal{A}| < \infty$, the weighted-degree kernel is defined by

$$\kappa(x, x') = \sum_{\ell=1}^{\ell_{\max}} \sum_{j=1}^{L-\ell+1} \mathbb{I}\{x[j]^\ell = x'[j]^\ell\}, \tag{1}$$

where $x[j]^\ell$ denotes the length-ℓ subsequence of x starting at position j and terminating at position $j + \ell - 1$. In a nutshell, it breaks x and x' into all possible subsequences up to a maximum length $\ell_{\max} \leq L$ and computes the number of matching subsequences. The WD-kernel SVM has been shown to achieve state-of-the-art prediction accuracies in many genomic discrimination tasks, including the detection of transcription start sites [38] and splice sites [37]—achieving the winning entry in the international comparison by [1] of 19 leading gene finders and remains still unbeaten. Efficient implementations such as the one contained in the SHOGUN machine-learning toolbox [33], which employs effective feature hashing techniques [36], have been applied to problems where millions of sequences, each with more than thousand positions, are processed at the same time [34].

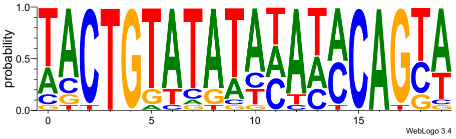


Fig. 1. Example of a *motif*, that is, an "interesting" subsequence in a sequence learning task that has a significance or impact on the label. The task here was gene detection and the motif has been generated using the *WebLogo 3* software [8]. The motif is illustrated as a *positional weight matrix* (PWM), where the size of a letter indicates the probability of its occurrence at a certain position in the motif. The likeliest entries are arranged top down.

Like many other kernels, the WD kernel is a black-box that hinders direct interpretation and analysis of the classifier that is output by the kernel-based learning algorithm (for other approaches for interpreting non-linear classification see e.g. [2,3,14,25,41]). It is an aim of this paper to work toward unveiling the function of such a classifier by computing the most important subsequences that determine the classifier's decision—the so-called *motifs*. A motif is a widespread and typical pattern in the input data that has, or is conjectured to have, a significance or impact on the associated label. For instance in the detection of gene starts, a motif is a nucleotide sequence (i.e., a string over the alphabet $\mathcal{A} = \{A, C, G, T\}$), which frequently appears at the start positions of genes in the DNA. For instance in Figure 1, we give an illustration of the motif `TACTGTATATATATACAGTA`.

The main contributions of this work can be summarized as follows:

1. Putting forward the work of [35] on positional oligomer importance matrices (POIMs), we propose a novel probabilistic framework to finally go the full way from the output of a WD-kernel SVM to the relevant motifs truly underlying the kernel machine's predictions.
2. To deal with the sheer exponentially large size of the feature space associated with the WD kernel, we propose a very efficient optimization framework based on advanced sequence decomposition techniques.
3. Our approach is able to even find multiple motifs consisting of hundreds of positions, while previous approaches are limited to either comparably short or contiguous motifs.
4. We demonstrate the efficiency and efficacy of our approach on synthetic data sets, on the USPS hand-written digits dataset, as well as on a *human* splice data set, where we achieve near-perfect motif reconstruction quality when evaluated by means of the JASPAR database [29].

2 Preliminaries

A first step towards the identification of motifs from the WD-kernel classifiers is achieved in [35], where the concept of positional oligomer importance matrices (POIMs) is introduced, which we review below, after giving more details on the concept of the WD kernel.

2.1 Weighted-Degree (WD) Kernel

The weighted-degree kernel is formally defined in (1). It is important to note, however, that we may equivalently represent the WD kernel by the corresponding binary feature embedding $\varPhi$, with $\kappa(x, x') = \langle \varPhi(x), \varPhi(x') \rangle$, where each entry of $\varPhi(x)$ represents a valid positional subsequence y of length $\ell \in \{1, \ldots, \ell_{\max}\}$ starting at position $j \in \{1, \ldots, L - \ell + 1\}$. A WD-kernel SVM then simply fits the parameter w of the linear model $s(x) := \langle w, \varPhi(x) \rangle$, which can, more concisely, be expressed as

$$s(x) = \sum_{\ell=1}^{\ell_{\max}} \sum_{i=1}^{L-\ell+1} w_{(x[i]^\ell, i)} \tag{2}$$

since $\varPhi(x)$ is inherently *sparse* (only the entries in $\varPhi(x)$ corresponding to the subsequences $y = x[i]^\ell$ with $\ell \in \{1, \ldots, \ell_{\max}\}$ and $i \in \{1, \ldots, L - \ell + 1\}$ are non-zero).

2.2 Positional Oligomer Importance Matrices (POIMs)

Given the base sequence length L, a *positional k-gram* is a subsequence $(y, j) \in \varSigma^k \times \{1, \ldots, L - k + 1\}$ of length k starting at a position j. *Positional oligomer* *importance matrices* (POIMs) assign each positional k-gram with an importance score. This allows us to visualize the significance of the various positional k-grams as illustrated in Fig. 2. To formally introduce the POIM approach, let Σ be a discrete alphabet, let $\mathcal{X} \sim \mathcal{U}(\Sigma^L)$ be a random variable that uniformly takes values in Σ^L, and let $x \in \Sigma^L$ be a realization thereof. For any positional k-gram (y, j) starting at position j, denote as

$$Q_{k,y,j} := \mathbb{E}[s(\mathcal{X})|\mathcal{X}[j]^k = y] - \mathbb{E}[s(\mathcal{X})], \tag{3}$$

the *POIM of order* k is defined as the tupel

$$Q \equiv Q_k := \left(Q_{k,y,j}\right)_{(y,j)\in\Sigma^k\times\{1,\ldots,L-k+1\}}.$$

We may interpret (3) as a measure for the contribution of the positional k-gram (y, j) to the SVM prediction function s as follows: a high value of $w_{(y,j)}$, by (2), implies a strong contribution to the prediction score $s(x)$ if and only if $y = x[j]^k$. We can very well visualize POIMs in terms of heatmaps as illustrated in Fig. 2, from which we may obtain the most discriminative features by manual inspection. As a first step towards a more automatic analysis of POIMs, [40] propose an extension of the POIM method, the so-called *differential POIM*, which aims to identify the most relevant motif lengths as well as the corresponding starting positions. Formally, the differential POIM Ω is defined as a $\ell_{\max} \times L$ matrix $\Omega := \left(\Omega_{\ell,j}\right)$ with entries

$$\Omega_{\ell,j} := \begin{cases} q_{\max}^{\ell,j} - \max\{q_{\max}^{\ell-1,j}, q_{\max}^{\ell-1,j+1}\} & \text{if } \ell \in \{2,\ldots,L\} \\ 0 \quad \text{elsewise}, \end{cases}$$

where $q_{\max}^{\ell,j} := \max_{y\in\Sigma^\ell} |Q_{\ell,y,j}|$. We can interpret $\Omega_{\ell,j}$ as an overall score for the general importance of the subsequence of length ℓ at position j.

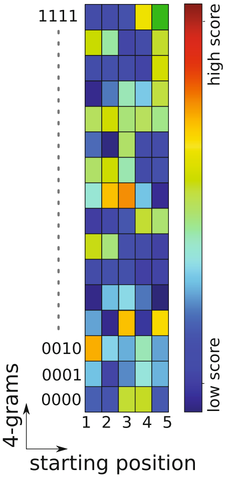


Fig. 2. Illustration of a POIM of k-grams ($k = 4$) over the binary alphabet $\mathcal{A} = \{0, 1\}$ and sequence length $L = 5$ for a trained kernel predictor. Each positional 4-gram corresponds to a cell, where the color indicates the significance of the positional 4-gram to the kernel predictor.

2.3 Shortcomings of POIMs

Although being a major step towards the explanation of trained WD kernel models, POIMs suffer from the fact that their size grows exponentially with the length of the motif, which renders their computation feasible only for rather small motif sizes, typically $k \leq 12$. It also hampers manual inspection (in order to determine candidate motifs) already for rather small motif sizes such as $k \approx 5$ and is prohibitive for $k \geq 10$. For example, a POIM of order $k = 5$ contains, at each position, already $4^5 \approx 1,000$ oligomers that a domain expert would have to manually inspect. Slightly increasing the motif length to $k = 10$ leads to an unfeasible amount of $4^{10} \approx 1,000,000$ subsequences per position in the POIM.

2.4 What is Coming Up: The Proposed Approach in a Nutshell

In this paper, we tackle obtaining motifs from a trained kernel machine via the use of POIMs from a different perspective. In a nutshell, our approach is the other way round (!): we propose a probabilistic framework to reconstruct, from a given motif, the POIM that is the most likely to be generated by the motif. By subsequently minimizing the reconstruction error with respect to the truly given POIM, we can in fact optimize over the motif in order to find the one that is the most likely to have generated the POIM at hand. The latter poses a substantial numerical challenge due to the extremely high dimensionality of the feature space. Figure 3 illustrates our approach.

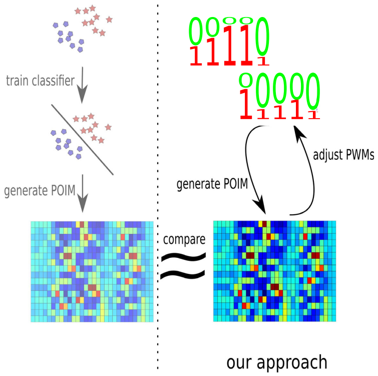


Fig. 3. Illustration of the proposed approach: extracting a motif (top right) from a trained kernel machine (top left) by approximating the corresponding POIM (bottom left) by another POIM (bottom right) that is derived from a set of candidate motifs, over which we optimize (top right).

3 Methodology for Revealing Discriminative Motifs by Mimicking POIMs

In this section, we introduce the proposed motifPOIM methodology for extraction of motifs from POIMs, state the optimization problem, and derive an efficient optimization procedure. In a nutshell, our motifPOIM methodolology (illustrated in Figure 3) is based on associating each candidate motif by a probability of occurrence at a certain location—which we call *probabilistic positional motif* (PPM)—and then (re-)construct from each PPM the POIM that is the most likely to be generated from the candidate PPM, which we call motifPOIM. The final motif is obtained by optimizing over the candidate motifs such that the reconstruction error of the motifPOIM with respect to the truly given POIM is minimized.

To this end, let us formally define the PPM as a tuple $m_k := (r, \mu, \sigma)$, where $r \in \mathbb{R}^{|\Sigma| \times k}$ and $\mu, \sigma \in \mathbb{R}$. We think of m_k as a candidate motif with PWM r and estimated starting position μ of which the variable σ encodes the uncertainty in the location of the motif. For this PPM we define a probabilistic model, with a probability of the starting position given by a Gaussian function with parameters μ and σ

$$P^1_{(z,i)}(m_k) := \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(i-\mu)^2}{2\sigma^2} \right),$$

and a probability for the motif sequence itself, given by the product of its PWM entries

$$P^2_{(z,i)}(m_k) := \prod_{\ell=1}^{k} r_{z_\ell, \ell} \, .$$

Under this probabilistic model, we define, in analogy to the SVM weight vector w occurring in (2), a motif weight vector $v \equiv v(m_k)$ with entries

$\big(v(m_k)\big)_{z,i} = v_{(z,i)}(m_k)$ defined as $v_{(z,i)}(m_k) := P^1_{(z,i)}(m_k)P^2_{(z,i)}(m_k)$, for any positional k-gram of length k, $(z,i) \in \Sigma^k \times \{1,\ldots,L-k+1\}$. Consequently, we define in analogy to (2) a function

$$\bar{s}(x|m_k) := \sum_{i=1}^{L-k+1} v_{(x[i]^k,i)}(m_k)\,. \tag{4}$$

By means of the above function, we can construct, from a PPM as defined in the paragraph above, a POIM $R \equiv R(m_k)$ with entries

$$R_{y,j}(m_k) := \mathbb{E}[\bar{s}(\mathcal{X}|m_k)|\mathcal{X}[j]^k = y] - \mathbb{E}[\bar{s}(\mathcal{X}|m_k)]\,. \tag{5}$$

Our overall aim is, by optimizing over the motifPOIM R, to approximate the original POIM (cf. also the illustration given by Figure 3). Due to the fact that searching for motifs of length k means computing POIMs of degree k, which is for longer PPMs ($k \geq 5$) computationally expensive, we have modified our optimization problem in a way that finding long PPMs can be accomplished using POIMs of lower degrees $\tilde{k} \in \{2,3\}$.The basic idea is to split longer PPMs of length k into shorter overlapping PPMs of length $\tilde{k} \leq k$ and use only the small POIM of degree $\tilde{k}$ for our optimization approach. First we define a set of smaller overlapping motifs, the SubPPMs, which should be devoted to the large PPM: A PPM of length k is modeled as a set of D SubPPMs, $D := k - \tilde{k} + 1$ with length $\tilde{k} \leq k$. The SubPPMs are defined by:

$$\tilde{m}_d(m_k, \tilde{k}) := (\tilde{r}, \tilde{\mu}, \sigma),\ \forall\ d = 0,\ldots,D-1$$

with $\tilde{\mu} := \mu + d$ and $\tilde{r} := r[d, d+\tilde{k}]$, where $r[d, d+\tilde{k}]$ is the d-th until the $(d+\tilde{k})$-th column of the PPMs PWM r.

3.1 Optimization Problem

We now derive the optimization problem for the extraction of motifs from POIMs. The core idea is to determine a motif m_k with an corresponding motif-POIM $R(m_k)$ that approximates the original POIM Q_k. To this end, let us introduce some notation. Let $\mathcal{K} \subset \mathbb{N}$ be the set of all motif lengths to be considered and $k_{\max} = \max\limits_{k\in\mathcal{K}} k$ the maximum length. The vector $T \in \mathbb{N}_0^{k_{\max}}$ contains the number of PPMs for each motif length, where T_k is the given number of PPMs of length k for all $k \in \mathcal{K}$. For example, when $\mathcal{K} = \{2,4,10\}$ and $T = (0,6,0,3,0,0,0,0,0,2)$, then the goal is to find 6 PPMs of length 2, 3 PPMs of length 4, and 2 PPMs of length 10. Our optimization method is as follows: given the set $\mathcal{K}$ and the vector T, we randomly initialize the PPMs $m_{k,t}$ $t = 1,\ldots,T_k$, $k \in \mathcal{K}$ and generate a set of motifPOIMs for the SubPPMs $\tilde{m}_d(m_k,\tilde{k})$, $d = 0,\ldots,D-1$. The optimization variables are the T_k many PPMs for all $k \in \mathcal{K}$. For obtaining the priorities of the PPMs we weight the PPMs by $\lambda_{k,t}$, $t = 1,\ldots,T_k$, $k \in \mathcal{K}$ and additionally optimize over the weights. Hence, the optimization variables are:

- PPM $m_{k,t} = (r_{k,t}, \mu_{k,t}, \sigma_{k,t})$, $\qquad t = 1,\ldots,T_k$, $k \in \mathcal{K}$,

$$\text{where } \mu_{k,t} \in \mathbb{R}, \sigma_{k,t} \in \mathbb{R}, r_{k,t} \in \mathbb{R}^{|\Sigma|\times k},\ t = 1,\ldots,T_k\,,\ k \in \mathcal{K}$$

– weight of $m_{k,t}$ $\qquad \lambda_{k,t} \in \mathbb{R}, \qquad t = 1, \ldots, T_k\,, \; k \in \mathcal{K}\,.$

A PPM generates a motifPOIM, which is given by the sum of D motifPOIMs generated by its SubPPMs. The sum of the weighted motifPOIMs, $\lambda_{k,t}R(m_{k,t}),\; t = 1, \ldots, T_k$, should estimate the POIM $Q_{\tilde{k}}$ for each $k \in \mathcal{K}$. The optimization problem is now that of minimizing the distance between the sum of the motifPOIMs and the original POIM, which leads to a non-convex optimization problem with the following objective function:

$$f(\eta) = \frac{1}{2} \sum_{k \in \mathcal{K}} \sum_{y \in \Sigma^{\tilde{k}}} \sum_{j=1}^{L} \left(\sum_{t=1}^{T_k} \lambda_{k,t} \sum_{d=0}^{D-1} R_{y,j}(\tilde{m}_d(m_{k,t}, \tilde{k})) - Q_{\tilde{k},y,j} \right)^2 , \tag{6}$$

where $\eta = (m_{k,t}, \lambda_{k,t}, \tilde{k})_{t=1,\ldots,T_k, k \in \mathcal{K}}$. The associated constrained non-linear optimization problem is thus as follows:

$$\begin{aligned} & \min_{(m_{k,t},\lambda_{k,t})_{t=1,\ldots,T_k,k\in\mathcal{K}}} \; f(\eta) && \qquad (7) \\ \text{subject to} \quad & \epsilon \le \sigma_{k,t} \le k, && t = 1, \ldots, T_k\,, \; k \in \mathcal{K} \\ & 1 \le \mu_{k,t} \le L - k + 1, && t = 1, \ldots, T_k\,, \; k \in \mathcal{K} \\ & 0 \le \lambda_{k,t} \le W, && t = 1, \ldots, T_k\,, \; k \in \mathcal{K} \\ & \epsilon \le r_{k,t,o,s} \le 1, && t = 1, \ldots, T_k\,, \; k \in \mathcal{K} \\ & && o = 1, \ldots, |\Sigma|, s = 1, \ldots, k\,, \sum_{o=1}^{|\Sigma|} r_{k,t,o,s} = 1 \end{aligned}$$

where $W \in \mathbb{R}^+$. The objective function $f(\eta)$ is defined on compact set U, since all parameters are defined in a closed and bounded, convex space. Consequently, if U is not empty, $f(\eta)$ is a continuously differentiable function, since its conforming parts, that is, the Gaussian function and the product of the PWM entries, all are continuously differentiable. Thus the global minimum of the optimization problem (7) is guaranteed to exist. Due to the non-convex nature of (7), however, there may exist multiple local minima.

3.2 Efficient Computation of motifPOIM

To allow for numerical optimization of (7), we need an efficient way of computing (5). To this end, note that (5) consists of two summands. The right-hand summand can be computed as follows:

$$\begin{aligned} \mathbb{E}[\bar{s}(\mathcal{X}|m_k)] &= \frac{1}{|\Sigma^L|} \sum_{x \in \Sigma^L} \bar{s}(x; m_k) = \frac{1}{|\Sigma^L|} \sum_{x \in \Sigma^L} \sum_{\ell=1}^{k} \sum_{i=1}^{L-\ell+1} v_{(x[i]^\ell, i)}(m_k) \\ &= \sum_{\ell=1}^{k} \sum_{i=1}^{L-\ell+1} \frac{1}{|\Sigma^L|} \sum_{x \in \Sigma^L} v_{(x[i]^\ell, i)}(m_k) = \sum_{\ell=1}^{k} \sum_{i=1}^{L-\ell+1} \frac{1}{|\Sigma^\ell|} \sum_{z \in \Sigma^\ell} v_{(z,i)}(m_k) \\ &= \sum_{\ell=1}^{k} \sum_{z \in \Sigma^\ell} \sum_{i=1}^{L-\ell+1} v_{(z,i)}(m_k) \mathbb{P}(\mathcal{X}[i]^\ell = z)\,. \end{aligned} \tag{8}$$

Furthermore, by an analogous computation, we compute the left-hand summand in (5) and obtain

$$\mathbb{E}[\bar{s}(\mathcal{X}|m_k)|\mathcal{X}[j]^k = y] = \sum_{\ell=1}^{k} \sum_{z \in \Sigma^\ell} \sum_{i=1}^{L-\ell+1} v_{(z,i)}(m_k)\mathbb{P}(\mathcal{X}[i]^\ell = z|\mathcal{X}[j]^k = y)\,. \tag{9}$$

We now consider this probability term and its influence on the summation in (5). To this end, we introduce the following notation as in [37].

Definition 1. *Two positional subsequences* (z,i) *and* (y,j) *of length* ℓ *and* k *are independent if and only if they do not share any position; in this case we write* $(y,j) \not\prec (z,i)$ *and* $(y,j) \prec (z,i)$ *otherwise (i.e., when they are dependent). If they are dependent and also agree on all shared positions we say they are* compatible *and we write* $(y,j) \precsim (z,i)$ *(and* $(y,j) \not\precsim (z,i)$ *if they are not compatible).*

According to the cases discussed in the above definition, the conditioned probability term can take the following values:

$$\mathbb{P}(\mathcal{X}[i]^\ell = z|\mathcal{X}[j]^k = y) = \begin{cases} \frac{1}{|\Sigma^\ell|} & \text{if } (y,j) \not\prec (z,i) \\ 0 & \text{if } (y,j) \not\precsim (z,i) \\ \frac{|\Sigma^c|}{|\Sigma^\ell|} & \text{if } (y,j) \precsim (z,i) \end{cases}\,, \tag{10}$$

where c is the number of shared and compatible positions of two positional subsequences:

$$c\big((y,j),(z,i)\big) = \begin{cases} \ell - |i-j| & \text{if } i < j \text{ and } (y,j) \precsim (z,i) \\ \ell & \text{if } i = j \text{ and } (y,j) \precsim (z,i) \\ k - |i-j| & \text{if } i > j \text{ and } (y,j) \precsim (z,i) \\ 0 & \text{else.} \end{cases}\,.$$

Taken the case $(y,j) \not\prec (z,i)$, the probability terms in the motifPOIM formula (5) subtract to zero, so that the positional subsequence (z,i) is not considered in the sum $R_{y,j}(m_k)$. Hence, in order to compute $R_{y,j}(m_k)$, it is sufficient to sum over two positional subsequence sets, where one contains all (z,i) with $(y,j) \precsim (z,i)$, $\mathcal{I}^{\precsim}_{(y,j)}$, and the others contains all (z,i) with $(y,j) \not\precsim (z,i)$, $\mathcal{I}^{\not\precsim}_{(y,j)}$:

$$R_{y,j}(m_k) = \sum_{(z,i)\in\mathcal{I}^{\precsim}_{(y,j)}} v_{(z,i)}(m_k)\Big(\frac{|\Sigma^c|}{|\Sigma^k|} - \frac{1}{|\Sigma^k|}\Big) + \sum_{(z,i)\in\mathcal{I}^{\not\precsim}_{(y,j)}} v_{(z,i)}(m_k)\Big(- \frac{1}{|\Sigma^k|}\Big)\Big)\,, \tag{11}$$

$$\text{where } \mathcal{I}^{\circ}_{(y,j)} := \Big\{(z,i) \in \Sigma^{|y|} \times \{1,\ldots,L-|y|+1\} | (y,j) \circ (z,i)\Big\} \text{ and } \circ \in \{\precsim, \not\precsim\}\,.$$

4 Empirical Analysis

In this section, we analyze our proposed mathematical model (7) empirically. After introducing the experimental setup, we evaluate our approach on the USPS data set, containing grayscale handwritten digit images. Afterwards, we conduct a biology experiment with a synthetic data set where we fully control the underlying ground truth. Finally, we investigate our model on a real *human* splice data set and compare our results to motifs contained in the JASPAR database [29]. As kernel-based learning algorithm, we use a support vector machine in all experiments.

4.1 Experimental Setup

For SVM training, we use the SHOGUN machine-learning toolbox [33]. The regularization constant C of the SVM and the degree d of the weighted-degree kernel are set to $C = 1$ and $d = 20$ for the biological experiments, which are proven default values. For the experiments on the USPS data, we set $d = 8$ and select C through model selection.

After SVM training, the POIM Q is generated through the Python script COMPUTE_POIMS.PY included in the SHOGUN toolbox. The Python framework obtains the trained SVM and the POIM of order k as parameters and returns the differential POIM and the regular POIMs $Q_l, l = 1, \dots, k_{poim}$. We set $k = 7$ because of memory requirements (storing all POIMs up to a degree of 10 requires about 4 gigabytes of space). Note that this is no restriction as our modified optimization problem (7) requires POIMs of degree two or three only. Nevertheless, POIMS of higher degree than three can provide additional useful information since they contain prior information about the optimization variables.

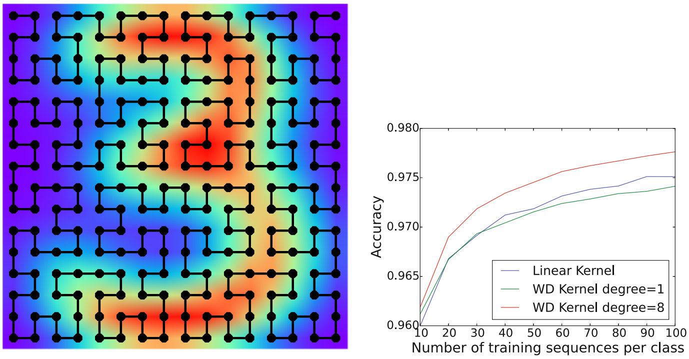
We then compute the differential POIM using the Python scripts included in the SHOGUN toolbox, where we search for points of accumulation of high scoring entries, from which we estimate the number of motifs as well as their length and starting position. Throughout the experiments, we use a greedy approach for estimating the initial values of PWMs given a POIM. Once the motif interval is estimated, we select the leading nucleotide from the highest scoring column entry within the interval from the corresponding POIM and initialize the respective PWM entry with a value of 0.7 and 0.1 for non-matches. Indeed, we found that this approach is more stable and reliable than using random initializations. These parameters serve as initialization for our non-convex optimization problem (7). To compute a PWM from the computed POIMs, we employ the L-BFGS-B Algorithm [23], where the parameters λ and σ are initialized as 1 and 0.01, respectively.

As a measure of the motif reconstruction quality (MRQ), we employ in the biological experiments the same score as in the established JASPAR SPLICE database [30]. Given a ground truth sequence motif t we test the reconstruction quality of an equally-sized, revealed motif r according to the following formula: $\text{MRQ} = \sum_{p=1}^{k} \left[\frac{1}{k} - \frac{1}{2k} \sum_{c \in \{A,C,G,T\}} (t_{cp} - r_{cp})^2 \right]$ We also introduce a second measure, the maximal-value MRQ (mvMRQ), which is defined in exactly the same way as the MRQ but uses the *maximum posteriori* motif $\hat{r} \in \{0,1\}^{4 \times k}$, that is, it considers only the most likely sequence in the motif, which can have the advantage of discarding potential noise in the data and motif.

4.2 Experimental Results for USPS Dataset

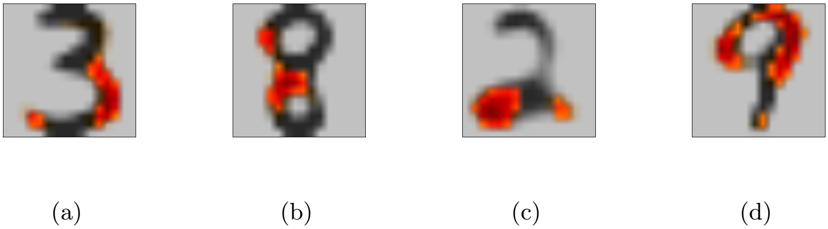
We first evaluate the proposed methodology on the USPS data set [15,16], which includes 9298 images of handwritten digits, encoded through gray scale values ranging in $[-1, 1]$. For pre-processing, the data was converted to a binary format by setting a threshold at -0.2 for the gray scale values. To preserve locality in the vectorial image representation, we further preprocessed the data by scanning the image using a Hilbert curve of order 4, which is a proven method for mapping images to sequences [6,9]. Fig. 4 (a) shows the path of the Hilbert-curve scan for the handwritten images of the digit three. To determine the justification of the use of a high-dimensional weighted degree kernel, we compare it with a linear kernel on the gray scale values as well as with the weighted degree kernel of degree one only. The results in terms of multi-class classification accuracy are shown in Fig 4 (b), where the SVM was trained in one-vs.-all scheme. We observe that a weighted degree kernel of degree 8 (dimensionality: $2^8 * 256 = 65536$) performs best in our experiments.

For the remaining experiments, we focus on the binary classification tasks of the handwritten digits three vs. eight and two vs. nine, respectively. These



(a) Hilbert Curve (b) SVM performance

Fig. 4. (a) Foreground: illustration of Hilbert-curve scanning (of order 4) of an image depicting of the handwritten digit three. The image is converted into a sequence through a curve that traverses the image in a way that mimics a fractal structure. It has been shown in [6,9] that this strategy is able to well capture the image's locality structure. The heatmap in the background shows the average feature values for the images of the digit three.
(b) The one-vs.-all SVM prediction accuracy is shown as a function of the number of training sequences per class for various WD sequence kernels over the Hilbert-scanned sequences and for a linear kernel on the gray-scale pixel values. The WD kernel of degree 8 performers best, even for only a small number of training sequences.



(a) (b) (c) (d)

Fig. 5. Illustration of the results found by our proposed framework, when training a WD-kernel SVM of degree 8 for the handwritten digits three vs. eight and two vs. nine, respectively. The highest scoring positions in the motif are highlighted in red. Note that these are very characteristic positions for the dissimilarities between both digits. The background the average feature values for the images of the respective digit.

respective digit pairs are considered to be especially difficult to discriminate. For both digit pairs we train a WD-kernel SVM of degree 8 on the Hilbert-scanned sequences. Afterward, we compute the POIM as described in Section 4.1 and use our presented methodology to find a motif that incorporates the discriminative positions of the SVM decision for both classes. In this experiment, we simply fix the length of the motif to 256, which thus coincides with the sequence length. The step of intializing the POIM parameter through analyzing the differential POIM is thus omitted in this experiment. The results, illustrated in Figure 5, show the precise coherence between the discriminative motifs found and the obvious individually characteristic differences of the two digits, respectively. For instance in the discriminative task three-vs.-eight, we can observe that the most distinctive positions in the motif of the digit eight (highlighted in red in Figure 5 (b)) are exactly the parts that are missing in the digit-three image.

4.3 Results for Synthetic Splice Site Experiments

Next, we evaluate the proposed methodology for biology DNA sequence data, by generating a synthetic data set, where we have full access to the underlying ground truth. This experiment aims at demonstrating the ability of our method in reconstructing the truly underlying motifs.

To this end, we generate the following sample sets: the sample set S_1 consists of 10,000 DNA sequences of length 30 over the alphabet $\{A, C, G, T\}^{30}$, randomly drawn from a uniform distribution $\mathcal{U}(\Sigma^L)$ over Σ^L. We subsequently modify 25% of the sequences by replacing the positions 6 to 11 by the synthetic target sequence CCTATA. These modified sequences form the positively labeled examples, while the remaining 75% of sequences are assigned with a negative label. The sample set S_2 includes the motif GATACATTAGGC of length 12 starting at position 16 in the positively labeled sequences. In the third sample set S_3 we insert both motifs at the same time.

The result of the realization of this synthetic experiments using the base sample S_1 and S_2 are shown in Figure 6. The corresponding motif/PWM computed by our approach correctly identifies the true underlying motif sequence as the most likely path in the PWM. More detailed results are shown in Table 1,

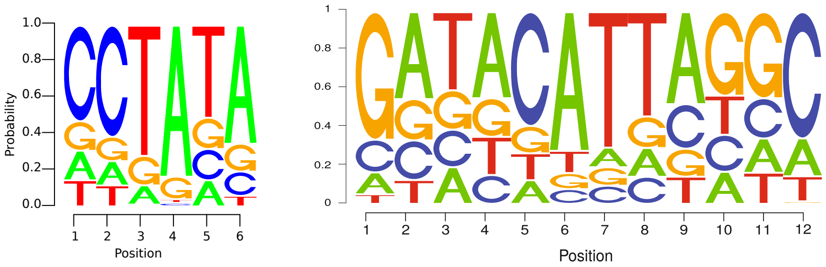


Fig. 6. Illustration of the motifs computed by our approach in the synthetic experiment: the size of a letter indicates the probability of occurrence of the corresponding nucleotide at a certain position in the motif. The left- and right-hand figures show the results for the synthetic data sets S_1 and S_2, respectively. Note that the truly underlying motifs where CCTATA for S_1 and GATACATTAGGC for S_2.

Table 1. Experimental results for the synthetic experiments on the three different sample sets S_1, S_2, and S_3.

sample set	SVM acc	#iter	time (s)	fevals	λ_{opt}	MRQ	mvMRQ
S_1	0.9987	157	13.2	116	1.0	0.93	1.0
S_2	1.0	31	19.7	64	1.0	0.65	1.0
S_3	1.0	31	25.87	64	0.42	0.85	1.0 (motif 1)
					0.58	0.84	1.0 (motif 2)

where, besides the MRQ and the mvMRQ value, we report also on the runtime of our approach, as well as the number of function evaluations, the optimal parameters for λ, the number of iterations needed, and the achieved SVM accuracy. Inspecting the mvMRQ, one can observe that even for the difficult dataset S_3, where we implanted both motives into the training sequences, we reconstruct both truly underlying motifs with 100% accuracy. The runtime of our approach ranges between 13 and 26 seconds.

4.4 Real-World Experiments on Human Splice Data

In this section, we evaluate our methodology on a *human* splice data set, which we downloaded from http://www.fml.tuebingen.mpg.de/raetsch/projects/lsmkl. For verifying our results we use the JASPAR database [29] (Available from http://jaspar.genereg.net), which provides us with a collection of important DNA motifs and also contains a splice site database. Note that real DNA sequences may contain non-polymorphic loci, which is why such a motif is not discriminative and we may thus not expect the SVM to identify this locus. We thus catch this special case and place this positional oligomer in the solution sequence. We apply the full experimental pipeline described in Section 4.1 to this data set.

Table 2. Execution times and optimal parameters for the *human* splice data set.

σ fixed	λ_{opt}	f_{opt}	time (s)	f evals	MRQ
0.01	0.005	78.97	37.91	24	90.1
0.1	0.84	59.48	28.3	20	97.58
1	1.67	57.18	33.53	17	97.03

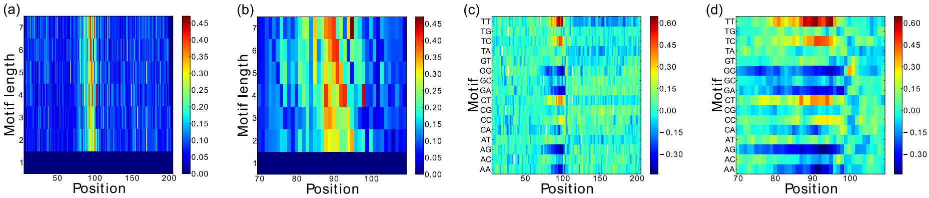


Fig. 7. Results of the real-world human splice experiment: Figures (a) and (c) show the differential POIM and the POIM of degree 2, respectively, for the entire sequence length of 200, while Figures (b) and (d) zoom into the "interesting" positions 70–110 only.

Figure 7 shows the preliminary results in terms of the differential POIM and the corresponding POIM of degree 2, shown for the entire sequence (see Figures 7 (a) and (c), respectively) as well as zoomed in for the "interesting" positions 70–110 of the sequence (see Figures 7 (b) and (d)). According to Figure 7 (b) the largest entry corresponds to a 7-mer that is found at position 95; furthermore, we observe high scoring entries for 7-,6- and 5-mers at position 85, from which we conclude that the discriminative motif starts at position 85 and ends at position 102. Thus, the motif we are searching is expected to have a length of 18 nucleotides, which we use as an initialization for our motifPOIM approach. We also account for non-polymorphic loci and find that the nucleotides A and G appear in all DNA sequences of the data set, always at the positions 100 and 101, respectively. We thus place them in the final PWM with a probability of 10%. The JASPAR splice database provides us with splice site motifs of length 20 only, which is why we search for motifs of the same size instead of the expected motif length 18.

The final results are shown in Figure 8, where the true underlying motif taken from the JASPAR splice database is shown in Figure (a), while the motif computed by our approach is shown in Figures (b)–(d). We observe a striking accordance with the true motif as evidenced by a high consensus score of 98.39

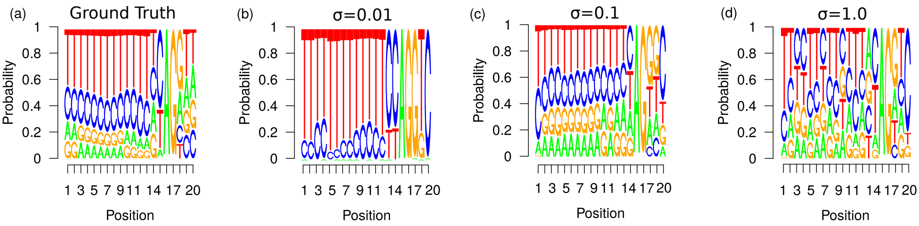


Fig. 8. Further results of the real-world human splice experiment: Figure (a) shows the (normalized) real splice sequence as taken from JASPAR. Figure (b)–(d) show the (normalized) computed PWMs for different values of the parameter σ. The best JASPAR score of 97.58, is achieved with $\sigma = 0.1$. This is, interestingly, followed by $\sigma = 1$ with a JASPAR score of 97.03 although the reconstructed motif of b) with $\sigma = 0.01$ and a score of 90.1 appears much more similar to the true motif in a).

for $\sigma = 0.1$, shown in Figure (c). Note that, for example, a completely random sequence (uniformly drawn nucleotides) has an average consensus of 89.31, which is greatly exceeded by our result. It is interesting to note that the function value corresponding to the best consensus score is suboptimal; this might indicate that the function is highly nonconvex with many local minima. Moreover, it is interesting to note that the PWM with the mixed nucleotides, shown in (d), is assigned a much higher accordance with the true motif than the well ordered one, shown in (b), which is more similar to the original JASPAR PWM. Furthermore, from Table 2, we observe moderate execution times of up to 32 seconds.

5 Conclusion and Discussion

Putting forward the work of [35] on positional oligomer importance matrices (POIMs), we have developed a new probabilistic methodology to automatically extract discriminative motifs from trained weighted-degree kernel machines such as support vector machines. To deal with the exponentially large size of the feature space associated with the SVM weight vector and the corresponding POIM ("[..] we realize that the list of POs can be prohibitively large for manual inspection." [35], page 8), we proposed an efficient optimization framework.

The results clearly illustrate the power of our approach in discovering discriminative motifs. For the experiment on handwritten digits, the proposed approach excels in finding intuitive motifs, as can be seen in Figure 5. In the synthetic experiments, the hidden motifs could be found and almost perfectly reconstructed. For the human splice site experiments, we recovered known motifs up to a very high precision of 98.39% as compared to the Jaspar splice data base.

We will provide the core algorithms as an add-on to the Python interface of the SHOGUN Machine Learning Toolbox. It is not only an established machine-learning framework, moreover, it already incorporates the possibility to extract positional-oligomer importance matrices (POIMs) of trained support vector machines using a WD-kernel. Ultimately, the usage by experimentalists will determine the utility of this approach and govern the direction of further extensions. A core issue might be the extension to other interesting kernels, such as, e.g., spectrum kernels [22], multiple kernels [17–19,21], other learning methods [11,12], or learning settings [10,20,39].

Acknowledgments. Acknowledgments MK acknowledges support by the German Research Foundation through the grant KL 2698/1-1 and KL 2698/2-1. MMCV and NG were supported by BMBF ALICE II grant 01IB15001B. We also acknowledges the support by the German Research Foundation through the grant DFG MU 987/6-1 and RA 1894/1-1. KRM thanks for partial funding by the National Research Foundation of Korea funded by the Ministry of Education, Science, and Technology in the BK21 program and the German Ministry for Education and Research as Berlin Big Data Center BBDC, funding mark 01IS14013A.

References

1. Abeel, T., de Peer, Y.V., Saeys, Y.: Towards a gold standard for promoter prediction evaluation. Bioinformatics (2009)
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLOS ONE (2015)
3. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Müller, K.R.: How to explain individual classification decisions. JMLR **11**, 1803–1831 (2010)
4. Ben-Hur, A., Ong, C.S., Sonnenburg, S., Schölkopf, B., Rätsch, G.: Support vector machines and kernels for computational biology. PLoS Comput Biology **4**(10), e1000173 (2008). http://www.ploscompbiol.org/article/info:doi/10.1371/journal.pcbi.1000173
5. Boser, B., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Haussler, D. (ed.) COLT. pp. 144–152. ACM (1992)
6. Chung, K.L., Huang, Y.L., Liu, Y.W.: Efficient algorithms for coding hilbert curve of arbitrary-sized image and application to window query. Information Sciences **177**(10), 2130–2151 (2007)
7. Cortes, C., Vapnik, V.: Support vector networks. Machine Learning **20**, 273–297 (1995)
8. Crooks, G., Hon, G., Chandonia, J., Brenner, S.: Weblogo: A sequence logo generator. Genome Research **14**, 1188–1190 (2004)
9. Dafner, R., Cohen-Or, D., Matias, Y.: Context-based space filling curves. In: Computer Graphics Forum, vol. 19, pp. 209–218. Wiley Online Library (2000)
10. Goernitz, N., Braun, M., Kloft, M.: Hidden markov anomaly detection. In: Proceedings of The 32nd International Conference on Machine Learning, pp. 1833–1842 (2015)
11. Görnitz, N., Kloft, M., Brefeld, U.: Active and semi-supervised data domain description. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009, Part I. LNCS, vol. 5781, pp. 407–422. Springer, Heidelberg (2009)
12. Görnitz, N., Kloft, M., Rieck, K., Brefeld, U.: Active learning for network intrusion detection. In: AISEC, p. 47. ACM Press (2009)
13. Görnitz, N., Kloft, M.M., Rieck, K., Brefeld, U.: Toward supervised anomaly detection. Journal of Artificial Intelligence Research (2013)
14. Hansen, K., Baehrens, D., Schroeter, T., Rupp, M., Müller, K.R.: Visual interpretation of kernel-based prediction models. Molecular Informatics **30**(9), September 2011. WILEY-VCH Verlag
15. Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., Tibshirani, R.: The elements of statistical learning, vol. 2. Springer (2009)
16. Hull, J.J.: A database for handwritten text recognition research. IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(5), 550–554 (1994)
17. Kloft, M., Brefeld, U., Sonnenburg, S., Zien, A.: lp-Norm Multiple Kernel Learning. JMLR **12**, 953–997 (2011)
18. Kloft, M., Brefeld, U., Düessel, P., Gehl, C., Laskov, P.: Automatic feature selection for anomaly detection. In: Proceedings of the 1st ACM Workshop on AISec, pp. 71–76. ACM (2008)
19. Kloft, M., Brefeld, U., Sonnenburg, S., Laskov, P., Müller, K.R., Zien, A.: Efficient and accurate lp-norm multiple kernel learning. Advances in Neural Information Processing Systems **22**(22), 997–1005 (2009)

20. Kloft, M., Laskov, P.: Online anomaly detection under adversarial impact. In: AISTATS, pp. 405–412 (2010)
21. Kloft, M., Rückert, U., Bartlett, P.: A unifying view of multiple kernel learning. Machine Learning and Knowledge Discovery in Databases pp. 66–81 (2010)
22. Leslie, C.S., Eskin, E., Noble, W.S.: The spectrum kernel: A string kernel for svm protein classification. In: Pacific Symposium on Biocomputing, pp. 566–575 (2002)
23. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**(3), 503–528 (1989). http://dx.doi.org/10.1007/BF01589116
24. Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of machine learning. MIT press (2012)
25. Montavon, G., Braun, M.L., Krueger, T., Müller, K.R.: Analyzing local structure in kernel-based learning: Explanation, complexity and reliability assessment. Signal Processing Magazine, IEEE **30**(4), 62–74 (2013)
26. Müller, K.R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks **12**(2), 181–201 (2001). http://dx.doi.org/10.1109/72.914517
27. Rätsch, G., Sonnenburg, S., Srinivasan, J., Witte, H., Müller, K.R., Sommer, R.J., Schölkopf, B.: Improving the caenorhabditis elegans genome annotation using machine learning. PLoS Comput. Biol. **3**(2), e20 (2007)
28. Rätsch, G., Sonnenburg, S.: Accurate splice site prediction for caenorhabditis elegans. Kernel Methods in Computational Biology, 277–298 (2004). MIT Press series on Computational Molecular Biology, MIT Press
29. Sandelin, A., Alkema, W., Engström, P., Wasserman, W.W., Lenhard, B.: Jaspar: an open-access database for eukaryotic transcription factor binding profiles. Nucleic Acids Research **32**(Database–Issue), 91–94 (2004)
30. Sandelin, A., Höglund, A., Lenhardd, B., Wasserman, W.W.: Integrated analysis of yeast regulatory sequences for biologically linked clusters of genes. Functional & Integrative Genomics **3**(3), 125–134 (2003)
31. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press, Cambridge (2002)
32. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation **10**(5), 1299–1319 (1998)
33. Sonnenburg, S., Rätsch, G., Henschel, S., Widmer, C., Behr, J., Zien, A., Bona, F.D., Binder, A., Gehl, C., Franc, V.: The SHOGUN machine learning toolbox. Journal of Machine Learning Research **11**, 1799–1802 (2010)
34. Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B.: Large scale multiple kernel learning. Journal of Machine Learning Research **7**, 1531–1565 (2006)
35. Sonnenburg, S., Zien, A., Philips, P., Rätsch, G.: POIMs: positional oligomer importance matrices – understanding support vector machine based signal detectors. Bioinformatics (2008). (received the Outstanding Student Paper Award at ISMB 2008)
36. Sonnenburg, S., Franc, V.: Coffin: a computational framework for linear SVMs. In: ICML, pp. 999–1006 (2010)
37. Sonnenburg, S., Schweikert, G., Philips, P., Behr, J., Rätsch, G.: Accurate Splice Site Prediction. BMC Bioinformatics, Special Issue from NIPS workshop on New Problems and Methods in Computational Biology Whistler, Canada, December 18, 2006, vol. 8(Suppl. 10), p. S7, December 2007
38. Sonnenburg, S., Zien, A., Rätsch, G.: ARTS: Accurate Recognition of Transcription Starts in Human. Bioinformatics **22**(14), e472–480 (2006)

39. Zeller, G., Goernitz, N., Kahles, A., Behr, J., Mudrakarta, P., Sonnenburg, S., Raetsch, G.: mtim: rapid and accurate transcript reconstruction from rna-seq data. arXiv preprint arXiv:1309.5211 (2013)
40. Zien, A., Philips, P., Sonnenburg, S.: Computing Positional Oligomer Importance Matrices (POIMs). Research Report; Electronic Publication 2, Fraunhofer Institute FIRST, December 2007
41. Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lengauer, T., Müller, K.R.: Engineering support vector machine kernels that recognize translation initiation sites in DNA. BioInformatics **16**(9), 799–807 (2000)