# Gradient-based Optimization of Kernel-Target Alignment for Sequence Kernels Applied to Bacterial Gene Start Detection

Christian Igel, *Senior Member, IEEE*, Tobias Glasmachers, Britta Mersch, Nico Pfeifer, and Peter Meinicke

**Abstract**

Biological data mining using kernel methods can be improved by a task-specific choice of the kernel function. Oligo kernels for genomic sequence analysis have proven to have a high discriminative power and to provide interpretable results. Oligo kernels that consider subsequences of different lengths can be combined and parameterized to increase their flexibility. For adapting these parameters efficiently, gradient-based optimization of the kernel-target alignment is proposed. The power of this new, general model selection procedure and the benefits of fitting kernels to problem classes are demonstrated by adapting oligo kernels for bacterial gene start detection.

**Index Terms**

sequence analysis, oligo kernel, translation initiation sites, model selection, kernel target alignment, support vector machines

## I. Introduction

Kernel-based learning algorithms have been successfully applied to a variety of sequence classification tasks within the field of bioinformatics [1]. Recently in [2], *oligo kernels* have been introduced for

Christian Igel is with the Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany, christian.igel@neuroinformatik.rub.de.

Tobias Glasmachers is with the Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany.

Britta Mersch is with the German Cancer Research Center, 69120 Heidelberg, Germany.

Nico Pfeifer is with the Institut für Informatik, Abteilung Simulation biologischer Systeme, Eberhard-Karls-Universität Tübingen, 72076 Tübingen, Germany.

Peter Meinicke is with the Institut für Mikrobiologie und Genetik, Abteilung für Bioinformatik, Georg-August-Universität Göttingen, 37077 Göttingen, Germany.

the analysis of biological sequence data, where the term oligo(mer) refers to short, single stranded DNA fragments. As shown in [2], decision functions based on oligo kernels are easy to interpret and to visualize. They can therefore be used to infer characteristic sequence features. In contrast to other approaches, oligo kernels allow for gradually controlling the level of position-dependency of the representation, that is, how important the exact position of an oligomer is. For example, measuring the similarity of two sequences by the standard Hamming distance is fully position-dependent (either two symbols at a given position are identical or not), whereas comparing just the frequencies of the symbols is completely position-independent (the position of a symbol within a sequence does not matter, just how often it occurs). The gradual control is a decisive feature compared to other string kernels for biological sequences, which usually provide either position-dependent [3] or completely position-independent representations [4], [5]. Measuring the similarity between sequences using kernels based on the edit distance between the sequences is an alternative approach in which the position-dependency can be controlled [6].

In this study, we look at *combined oligo kernels* [2], which consider oligomers of different lengths. This kernel allows to control the position-dependency for each oligomer length individually and can therefore be better adapted to a particular prediction or data mining problem. This leads us to one of the key problems of all kernel-based methods, namely *model selection*, that is, finding the appropriate kernel for a given task. Typically, a parameterized family of kernel functions is considered and model selection reduces to real-valued parameter optimization. When using the combined oligo kernel for biological sequence analysis, we want to adapt the parameters that control position-dependency of oligomers of a particular length. The most sophisticated algorithms for model selection are gradient-based methods [7], [8], [9], [10], [11], [12]. However, they require the definition of a differentiable criterion for the performance of a kernel. Recently, the *kernel-target alignment* has been proposed as a criterion for kernel adaptation [13], [14], [15]. In this study, we derive gradient-based optimization of the kernel-target alignment leading to a general, efficient model selection method applicable to multiple kernel parameters.

This new model selection method enables us to adjust the combined oligo kernel for a given task. The power of this approach is demonstrated by applying it to the design of support vector machines [16] for the prediction of bacterial gene starts in genomic sequences [17]. Although exact localization of gene starts is crucial for correct annotation of bacterial genomes, it is difficult to achieve with conventional gene finders, which are usually restricted to the identification of long coding regions. The prediction of gene starts therefore provides a biologically relevant signal detection task, which has been successfully

approached by machine learning algorithms and is well-suited for the evaluation of our kernel optimization scheme.

In the following, we derive gradient-based optimization of the kernel target alignment for model selection. In Section III the oligo kernel is introduced, and in Section IV experiments using optimized oligo kernels for bacterial gene start prediction are presented.

## II. Kernel Selection Using Gradient-based Optimization of the Kernel-Target Alignment

The basic idea of kernel methods for classification is to map the input patterns (here biological sequences) to a feature space endowed with a dot product and to classify the patterns in the feature space using a well-understood algorithm in which all operations in the feature space can be expressed by dot products. The trick is to compute these inner products efficiently in the input space using a kernel function. Choosing the right kernel and thereby the right feature space is the most important aspect when designing a kernel classifier.

In this section, we first briefly describe support vector machines, the most prominent kernel-based learning method. Then we present the kernel-target alignment as a criterion of how well a kernel fits a certain data set. The gradient of the kernel-target alignment is derived, which can be used to select appropriate kernel parameters for a given problem.

### A. Support Vector Machines

In this study, we consider $L_1$-norm *soft margin support vector machines* (SVMs) for binary classification [16]. Let $(x_i, y_i)$, $1 \leq i \leq \ell$, be *consistent* training examples, where $y_i \in \{-1, 1\}$ is the label associated with input pattern $x_i \in \mathscr{X}$. The main idea of SVMs is to map the input patterns to a feature space $\mathscr{F}$ and to separate the transformed data linearly in $\mathscr{F}$.

The transformation $\phi : \mathscr{X} \to \mathscr{F}$ is implicitly done by a kernel $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$, which computes a scalar (inner) product in the feature space efficiently, that is, $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$. The kernel function $k$ has to be *positive definite*, that is, for every finite set of patterns $x_i$, $1 \leq i \leq \ell$, the matrix $G \in \mathbb{R}^{\ell \times \ell}$ with $G_{ij} = k(x_i, x_j)$ has to be positive definite (i.e., $a'Ga \geq 0$ for all $a \in \mathbb{R}^\ell$, we use the term *strictly positive definite* if strict inequality is required for non-zero $a$). Patterns are classified by the sign of a function $f$

of the form

$$f(x) = \langle w, \phi(x) \rangle + b = \sum_{i=1}^{\ell} \alpha_i^* y_i k(x_i, x) + b \ . \tag{1}$$

The real-valued coefficients $\alpha_i^*$ defining the weight vector $w = \sum_{i=1}^{\ell} y_i \alpha_i^* \phi(x_i)$ and $b$ are determined by solving the following quadratic optimization problem

$$\min_{w,b} \quad H[f] = \sum_{i=1}^{\ell} [1 - y_i f(x_i)]_+ + \frac{1}{2C} \|w\|^2 \ , \tag{2}$$

where $[z]_+ = 0$ if $z < 0$ and $[z]_+ = z$ otherwise. The first part penalizes patterns that are not classified correctly with a particular *margin* (i.e., distance from the separating hyperplane in $\mathscr{F}$). The second part regularizes the solution, in the sense that minimizing the norm of the weight vector corresponds to minimizing the norm of the function $\tilde{f}(x) = \langle w, \phi(x) \rangle$ in $\mathscr{F}$. If $\sum_{i=1}^{\ell} [1 - y_i f(x_i)]_+ = 0$, minimizing $\|w\|$ corresponds to maximizing the minimum distance of a training pattern from the separating hyperplane in $\mathscr{F}$. The *regularization parameter* $C$ controls the trade-off between the two parts of the objective function.

In practice, the coefficients $\alpha_i^*$ are computed by maximizing the *dual optimization problem*

$$E[\alpha] = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \tag{3}$$

subject to $\sum_{i=1}^{\ell} \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$ for $i = 1, \ldots, \ell$. The optimal value for $b$ can then be determined based on the solution $\alpha^* \in \mathbb{R}^{\ell}$. The patterns $x_i$ with $\alpha_i > 0$ are called *support vectors*. For solving the dual quadratic optimization problem we use a *sequential minimal optimization* (SMO, cf. [18]) approach based on second order information as proposed in [19], [20].

For an introduction to SVMs we refer to the literature (e.g., [21] or the textbooks [22], [23], [24]).

### *B. Model Selection and Kernel-Target Alignment*

The right choice of a kernel function, which implicitly determines the feature space $\mathscr{F}$, is crucial for the performance of the learning machine. Choosing an appropriate kernel and thereby defining a metric between input patterns that fosters correct classification is the model selection problem in the context of kernel-based methods. Usually, a parameterized family of kernel functions is considered. In this case model selection reduces to real-valued parameter optimization. Still, it is necessary to pick an appropriate family of kernel functions to choose from, a performance measure (i.e., a heuristic to compare kernels by quantifying how well they are suited for the problem class at hand), and an optimization strategy. If the kernel space has a differentiable structure and the performance measure is differentiable, the optimization

methods of choice for adapting multiple hyperparameters are iterative, gradient-based approaches. If these assumptions are not met, direct search methods such as grid-search, which is only applicable in case of very few parameters, or evolutionary algorithms [25], [26] are used.

Usually, gradient-based approaches rely on performance measures based on *radius-margin bounds* [7], [8], [9], [10], [11], [12]. In each iteration, radius-margin performance criteria require the training of the learning machine and the solution of an additional quadratic program to compute the radius of the smallest ball enclosing the training data in feature space. Here, we consider a different criterion for model selection, the kernel-target alignment [13], [14], [15]. It can be calculated efficiently, independently from the actual learning algorithm, makes use of the information from the complete training data set—and it is differentiable.

We consider a consistent training data set comprising $\ell$ training patterns $x_i \in \mathscr{X}$ with labels $y_i \in \{-1,+1\}$. On this set every positive definite kernel function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ defines a symmetric positive definite *kernel matrix* (*Gram matrix*) $G \in \mathbb{R}^{\ell\times\ell}$ by $G_{ij} := k(x_i,x_j)$. On the training set we can measure the similarity of two kernel functions $k_1$ and $k_2$ by the normalized inner product (i.e., the cosine of the angle)

$$S(k_1,k_2) := \frac{\langle G_{k_1}, G_{k_2}\rangle}{\sqrt{\langle G_{k_1}, G_{k_1}\rangle\langle G_{k_2}, G_{k_2}\rangle}} \tag{4}$$

between the corresponding kernel matrices $G_{k_1}$ and $G_{k_2}$, where the inner product between matrices is defined by $\langle A,B\rangle := \sum_{n,m=1}^{\ell} A_{nm}B_{nm}$ for $A,B \in \mathbb{R}^{\ell\times\ell}$.

We now consider the function

$$\overline{y} : \mathscr{X} \to \mathbb{R} \;, \qquad x \mapsto \begin{cases} y_m & \text{if } x = x_m \\ 0 & \text{otherwise} \end{cases} , \tag{5}$$

which assigns the observed label to every input pattern in the training set and assigns zero ("don't know") to every unseen input pattern. Let $y = (y_1,\ldots,y_\ell)'$. The outer product $yy'$ defines a positive definite rank one matrix with $(yy')_{ij} = y_iy_j$. It is the kernel matrix of the kernel function $\overline{yy}(x,z) := \overline{y}(x)\cdot\overline{y}(z)$ which can be thought of as an *empirical kernel* build of the training data. Obviously, it perfectly suits the training data. This observation leads to the definition of the *kernel-target alignment*

$$\hat{A}(k) := S(k,\overline{yy}) = \frac{\langle G, yy'\rangle}{\ell\sqrt{\langle G,G\rangle}} = \frac{\sum_{i,j=1}^{\ell} y_iy_jk(x_i,x_j)}{\ell\sqrt{\sum_{i,j=1}^{\ell} k(x_i,x_j)^2}} \;. \tag{6}$$

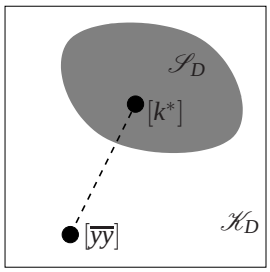


Fig. 1. Schema of model selection based on kernel-target alignment. The goal is to find a kernel $k^*$ within a restricted family of "reasonable" kernel functions such that the normalized Gram matrix induced by $k^*$ has the smallest distance to the normalized empirical kernel matrix $yy'$, see Appendix B for a more formal treatment of this topic and a precise definition of the labels.

Without normalization, kernel-target alignment corresponds to *kernel polarization*, which was proposed recently for model selection [27].

The kernel-target alignment measures the similarity of the kernel with $\overline{yy}$ on the observed data. We can rewrite $\langle G, yy' \rangle = \sum_{y_i=y_j} k(x_i,x_j) - \sum_{y_i \neq y_j} k(x_i,x_j)$. Thus, $\langle G, yy' \rangle$ (and therefore $\hat{A}(k)$ if only normalized Gram matrices are considered) is large if the similarity measure induced by the kernel is large for input patterns of the same class and small for patterns from different classes. This is the intuitive idea behind preferring a kernel with high kernel-target alignment, because the alignment reflects how well induced similarity measure and class membership match.

A model selection strategy based on the kernel-target alignment (and kernel polarization) requires a considerate choice of the kernel family, guided by prior knowledge about the problem domain. The kernel-target alignment is maximized by the empirical kernel, which is of course an undesired solution of the model selection problem. Thus, the empirical kernel must not be an element of the family of functions from which the kernel is selected. A schema and an additional geometric interpretation of model selection using the kernel-target alignment is provided in Fig. 1 and in Appendix B. It is important to stress that maximization of the kernel-target alignment does not aim at generalization properties of some classifier. The same holds for kernel polarization. Both measures are maximized if a kernel reflects the properties of the training data set used to define the empirical kernel. In order to prevent overfitting, only parameters that control general properties of the kernel family and that do not allow adaptation to individual input patterns should be optimized using these criteria.

It is a decisive feature of optimizing the kernel-target alignment for model selection that it is independent of the actual learning machine. No computationally expensive training of a classifier is necessary in the model selection process. Further, the resulting kernel can be plugged into different learning machines.

However, this lack of specificity can also be viewed as one of the main drawbacks of the approach, as the optimal feature space representation surely depends on the classification algorithm.

### C. Gradient of Kernel-Target Alignment

We propose to optimize the kernel-target alignment using gradient-based algorithms. The partial derivative of the kernel-target alignment with respect to a parameter $h$ of the kernel $k$ with corresponding Gram matrix $G$ is given by

$$\frac{\partial \hat{A}}{\partial h}(k) = \frac{\langle \frac{\partial G}{\partial h}, yy' \rangle \cdot \langle G, G \rangle - \langle G, yy' \rangle \cdot \langle \frac{\partial G}{\partial h}, G \rangle}{\ell \langle G, G \rangle^{3/2}} \tag{7}$$

using $\langle \frac{\partial G}{\partial h}, B \rangle = \sum_{n,m=1}^{\ell} \frac{\partial G_{nm}}{\partial h} B_{nm} = \sum_{n,m=1}^{\ell} \frac{\partial k(\mathbf{s}_n, \mathbf{s}_m)}{\partial h} B_{nm}$.

To the best of our knowledge, standard kernel-target alignment has neither been combined with efficient gradient-based optimization techniques nor applied to complex string kernels so far. Nevertheless, kernel-target alignment and kernel polarization have already proven to be well suited for model selection (e.g., see [13], [27]). In [11] a related, but more complex criterion is suggested for model selection. This measure is optimized by simple gradient-descent and discussed in the context of the kernel-target alignment. However, when we used the criterion proposed in [11] instead of $\hat{A}$ for adapting trimer and combined oligo kernels in our experiments described in section IV, the model selection led to degenerate kernels resulting in very poor performance.

## III. Oligo Kernels for Sequence Analysis

In this section, oligo kernels for the analysis of biological sequence data are described. These kernels have a high discriminative power and yield classifiers that are easy to interpret and to visualize [2]. The gradient of the *combined oligo kernel* with respect to its hyperparameters is derived for gradient-based model selection.

### A. Oligo kernels

The feature space representation induced by oligo kernels can be described in terms of *oligo functions* [2], which encode occurrences of oligomers in sequences with an adjustable degree of positional uncertainty. We consider finite sequences over an alphabet $\mathscr{A}$. In our context, subsequences $\omega \in \mathscr{A}^K$ of length $K$ are called *K-mers* (i.e., oligomers of length $K$). For a sequence $\mathbf{s}$ containing the $K$-mer $\omega \in \mathscr{A}^K$ at

positions $S^{\mathbf{s}}_{\omega} = \{p_1, p_2, \dots\}$, the oligo function is given by

$$\mu_{\omega}(t) = \sum_{p \in S^{\mathbf{s}}_{\omega}} \exp\left(-\frac{1}{2\sigma_K^2}(t-p)^2\right) \quad , \tag{8}$$

see Fig. 2 for an example. The continuous position variable $t$ is not restricted to a discrete domain so far. The *smoothing parameter* $\sigma_K$ adjusts the width of the Gaussians centered on the observed oligomer positions and determines the degree of position-dependency of the feature space representation. While small values for $\sigma_K$ imply peaky functions, large values imply flatter functions.

For a sequence $\mathbf{s}$ the occurrences of all $K$-mers contained in $\mathscr{A}^K = \{\omega_1, \omega_2, \dots, \omega_m\}$ can be represented by a vector of $m$ oligo functions. This yields the final feature space representation $\Phi^K(\mathbf{s}) = [\mu_{\omega_1}, \mu_{\omega_2}, \dots, \mu_{\omega_m}]'$ of that sequence. The feature space objects are vector-valued functions. This can be stressed using the notation

$$\phi^K_{\mathbf{s}}(t) = [\mu_{\omega_1}(t), \mu_{\omega_2}(t), \dots, \mu_{\omega_m}(t)]' \quad . \tag{9}$$

Each component corresponds to the oligo function of a particular $K$-mer. This representation is well-suited for the interpretation of discriminant functions and visualization [2]. To make it practical for learning, we construct a kernel function to compute the dot product in the feature space efficiently. The inner product of two sequence representations $\phi^K_i$ and $\phi^K_j$, corresponding to $k_K(\mathbf{s}_i, \mathbf{s}_j)$, is given by

$$\begin{aligned}\left\langle \phi^K_i, \phi^K_j \right\rangle := \int \phi^K_i(t) \cdot \phi^K_j(t) \mathrm{d}t &= \sum_{\omega \in \mathscr{A}^K} \sum_{p \in S^i_{\omega}} \sum_{q \in S^j_{\omega}} \int \exp\left(-\frac{(t-p)^2}{2\sigma_K^2}\right) \exp\left(-\frac{(t-q)^2}{2\sigma_K^2}\right) \mathrm{d}t \\ &\propto \sum_{\omega \in \mathscr{A}^K} \sum_{p \in S^i_{\omega}} \sum_{q \in S^j_{\omega}} \exp\left(-\frac{1}{4\sigma_K^2}(p-q)^2\right) := k_K(\mathbf{s}_i, \mathbf{s}_j)\end{aligned} \tag{10}$$

using $\phi_i := \phi_{\mathbf{s}_i}$ and $S^i_{\omega} := S^{\mathbf{s}_i}_{\omega}$. In Appendix A it is shown that oligo kernels are valid positive definite kernels.

The feature space representations of two sequences may have different norms. In order to improve comparability between sequences of different lengths, we compute the *normalized oligo kernel*

$$\tilde{k}_K(\mathbf{s}_i, \mathbf{s}_j) = \frac{k_K(\mathbf{s}_i, \mathbf{s}_j)}{\sqrt{k_K(\mathbf{s}_i, \mathbf{s}_i) k_K(\mathbf{s}_j, \mathbf{s}_j)}} \quad . \tag{11}$$

From the above definition of the oligo kernel, it is easy to see the effect of the smoothing parameter $\sigma_K$, see also Fig. 2. For the limiting case $\sigma_K \to 0$ with no positional uncertainty only oligomers that occur at the same positions in both sequences contribute to the sum. In general it is not appropriate to represent oligomer occurrences without positional uncertainty. This would imply zero similarity between two sequences if no $K$-mer appears at *exactly* the same position in both sequences. Regarding the other

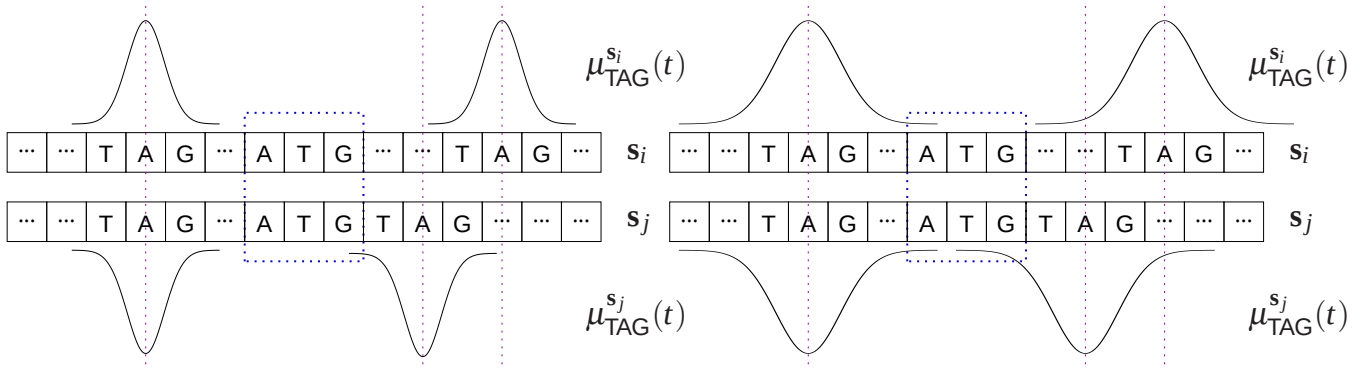


Fig. 2. Example of two sequences $\mathbf{s}_i$ and $\mathbf{s}_j$ and the corresponding oligo functions for $\omega = \mathsf{TAG}$ for small (left) and large (right) smoothing parameter $\sigma_3$.

extreme with maximum positional uncertainty, for $\sigma_K \to \infty$ position-dependency of the kernel completely vanishes. In this case, all terms of oligomers, occurring in both sequences, contribute equally to the sum, regardless of their distance and the oligo kernel becomes identical to the spectrum kernel [4].

It is beneficial to consider oligomers of different lengths. In [2], the *combined oligo kernel*

$$\tilde{k}_{\kappa\text{-combined}}(\mathbf{s}_1, \mathbf{s}_2) = \frac{1}{\kappa} \sum_{i=1}^{\kappa} \tilde{k}_i(\mathbf{s}_1, \mathbf{s}_2) \tag{12}$$

with $\kappa = 6$ and individual values for $\sigma_1, \ldots, \sigma_6$ was introduced. The individual smoothing parameters $\sigma_i$ allow for different degrees of position-dependency for $K$-mers depending on their length $K$. For example, the kernel can be adjusted in a way that matching trimers have to be at almost the same position for two sequences to be considered similar, whereas at the level of pentamers just their frequency matters.

The learning machines using $\tilde{k}_{6\text{-combined}}$ performed better than the machines using a single oligo kernel $\tilde{k}_i(\mathbf{s}_1, \mathbf{s}_2), i = 1, \ldots, 6$ in [2]. Because grid-search for six parameters is prohibitive, the smoothing parameters were tuned by considering each of the six kernels $\tilde{k}_i$ separately. Here we use gradient-based optimization of the kernel-target alignment for adjusting these hyperparameters simultaneously.

### *B. Motif Oligo Kernels*

Although adapted oligo kernels already show very good classification performance as demonstrated in the following section, in practice one would use these kernels in combination with *motif oligo kernels*, additional biological a priori information, and perhaps even other classification tools. Oligo kernels need not be defined over all oligomers of or up to a certain length. The kernel can also be defined over an arbitrary finite set $A_{\text{motifs}} \subset \cup_{k=1}^{\infty} \mathscr{A}^k$ of sequences of different lengths, in particular over a set of sequence

motifs relevant for the prediction task at hand. A set of motifs $A_{\text{motifs}} = \{\omega_1, \ldots, \omega_m\}$ leads straightforward to a feature space representation $\Phi^{A_{\text{motifs}}}(\mathbf{s}) = [\mu_{\omega_1}, \mu_{\omega_2}, \ldots, \mu_{\omega_m}]'$ of a sequence $\mathbf{s}$ and corresponding motif oligo kernels $k_{A_{\text{motifs}}}$ and $\tilde{k}_{A_{\text{motifs}}}$, which can additively be combined with the standard oligo kernel. Similarly, the oligo kernel can be coupled with kernel functions considering additional properties beyond the sequence information. However, in the following we consider only standard combined oligo kernels. This allows for a fair comparison with other sequence kernels and makes the improvements achieved by our model selection approach, which adapts kernels to specific problems, directly visible.

### C. Gradient of Combined Oligo Kernels

For gradient-based adaptation of the kernel parameters, we compute the partial derivatives of the combined oligo kernel with respect to its hyperparameters. For a smoothing parameter $\sigma_i$ we get

$$\frac{\partial \tilde{k}_{\kappa\text{-combined}}}{\partial \sigma_i}(\mathbf{s}_1, \mathbf{s}_2) = \frac{\frac{\partial k_i}{\partial \sigma_i}(\mathbf{s}_1, \mathbf{s}_2) k_i(\mathbf{s}_1, \mathbf{s}_1) k_i(\mathbf{s}_2, \mathbf{s}_2) - \frac{k_i(\mathbf{s}_1, \mathbf{s}_2)}{2} \left[ \frac{\partial k_i}{\partial \sigma_i}(\mathbf{s}_1, \mathbf{s}_1) k_i(\mathbf{s}_2, \mathbf{s}_2) + \frac{\partial k_i}{\partial \sigma_i}(\mathbf{s}_2, \mathbf{s}_2) k_i(\mathbf{s}_1, \mathbf{s}_1) \right]}{\kappa \sqrt{k_i(\mathbf{s}_1, \mathbf{s}_1) k_i(\mathbf{s}_2, \mathbf{s}_2)}^{3/2}} \tag{13}$$

and

$$\frac{\partial k_i}{\partial \sigma_i}(\mathbf{s}_1, \mathbf{s}_2) = \sum_{\omega \in \mathscr{A}^K} \sum_{p \in S^1_\omega} \sum_{q \in S^2_\omega} \frac{1}{2\sigma_i^3} (p-q)^2 \cdot \exp\left( -\frac{1}{4\sigma_i^2} (p-q)^2 \right) \quad . \tag{14}$$

Combining this result with the gradient of the kernel-target alignment $\hat{A}$ derived in Section II-C allows us to perform gradient-based optimization in the space of oligo kernels following $\partial \hat{A} / \partial \sigma_i$, $1 \leq i \leq \kappa$.

## IV. Experiments

We apply 1-norm soft margin SVMs as described in Section II-A with optimized combined oligo kernels to the detection of bacterial gene starts [17]. First the problem is outlined. Then we concisely describe the *locality improved kernel* [1], [28] and simple *Markov chain models* [29], which we consider for comparison. After that we give details about the model selection process. Finally the experimental results are presented.

### A. Problem Description

To extract protein-encoding sequences from nucleotide sequences is an important task in bioinformatics. For this purpose it is necessary to detect locations at which coding regions start. These locations are called *translation initiation sites* (TIS).

We consider (sense strand) DNA sequences, that is, strings over the alphabet $\mathscr{A} = \{\mathsf{A}, \mathsf{T}, \mathsf{C}, \mathsf{G}\}$. A TIS contains the start codon (substring) ATG or rarely GTG or TTG (in our example there is only one known case where also ATT serves as a start codon). The start codon marks the position at which the translation starts. Not every ATG triplet is a start codon, even if it is the first one on the transcribed mRNA when scanning starting from the 5' end. Therefore it must be decided whether a particular ATG corresponds to a start codon or not. This classification problem can be solved automatically using machine learning techniques, in which usually the neighborhood of nucleotides around potential TISs, probably combined with additional features, is used as input pattern to a classifier. Various successful applications of established statistical methods and computational intelligence techniques have been reported (e.g., [30], [31], [32], [28], [2], [6], [33], [34], [35]). *Markov chain models* (see section IV-C) were first used by Salzberg [30], *neural networks* were used for example in [31], [32], and SVMs in [28], [2], [6], [35]. In [33], Markov models and neural networks were combined. In addition to these supervised learning approaches, unsupervised methods have recently been applied to the problem of TIS prediction. In particular, Tech and Meinicke [34] iterate a process of supervised model building and reassigning input patterns to the classes of positive and negative examples. This algorithms yields very good results, which still depend on the quality of the supervised model building. Thus, improving the supervised putative TIS classification would also improve the unsupervised algorithm described in [34], which relies on Markov chain models with *positional smoothing*.

When discussing TIS detection, we have to distinguish between eukaryotes and prokaryotes, that is, between organisms in which the genetic material is organized into membrane-bound nuclei and organisms without a cell nucleus. In contrast to prediction of eukaryotic TIS (e.g., in [30], [31], [32], [28], [6]) there is no biological justification for using a general learning machine across different species for prediction of prokaryotic TIS. For this reason, learning of prokaryotic TISs is always restricted to a limited amount of species-specific examples and model selection methods have to cope with small data sets.

We chose an experimental setup to simulate a later step in a reannotation process, where a subset of all TISs in a prokaryotic genome has been verified on the basis of biological knowledge. These verified translation starts can be used to build a TIS classifier which in turn can be applied to correct or verify the putative TIS locations of the remaining genes of the genome, which have been found by a conventional tool for detection of open reading frames of a significant length (e.g., see [36]).

To create a reliable data set, we selected *E. coli* genes from the EcoGene database [37] and considered

only those entries with biochemically verified N-terminus. The neighboring nucleotides were looked up in the GenBank file U00096.gbk [38]. From the 732 positive examples (i.e., we have to deal with small data sets compared to the analysis of eukaryotic sequence databases) we created associated negative examples. For the negative examples we extracted sequences centered around a codon from the set $\{\mathsf{ATG}, \mathsf{GTG}, \mathsf{TTG}\}$ and accepted them if the codon is in-frame with one of the appropriate start sites used as a positive case, its distance from a real TIS is less than 80 nucleotides, and no in-frame stop codon occurs in between. This data selection generates a difficult benchmark because the negative TISs in the dataset are both in-frame with and in the neighborhood of the real TIS.

We finally obtained a set of 1248 negative examples. The length of each sequence is 50 nucleotides, with 32 located upstream and 18 downstream including the start codon.

To minimize random effects, we generated 50 different partitionings of the data into training and test sets. Each training set contained 400 sequences plus the associated negatives, the corresponding test set 332 sequences plus the associated negatives. The data sets can be obtained from:
http://www.neuroinformatik.rub.de/PEOPLE/igel/data/TIS-50.tgz

### *B. Locality Improved Kernel*

For comparison, we consider the locality improved kernel [1], [28]. It counts matching nucleotides and considers local correlations within local windows of length $2l+1$. Given two sequences $\mathbf{s}_i$, $\mathbf{s}_j$ of length $L$ the locality improved kernel is given by

$$k_{\text{locality}}(\mathbf{s}_i, \mathbf{s}_j) = \sum_{p=1}^{L} \left( \sum_{t=\max(1,p-l)}^{\min(L,p+l)} v_{t+l-p} \cdot \text{match}_t(\mathbf{s}_i, \mathbf{s}_j) \right)^d \tag{15}$$

with $\text{match}_t(\mathbf{s}_i, \mathbf{s}_j)$ equal to one if $\mathbf{s}_i$ and $\mathbf{s}_j$ have the same nucleotide at position $t$ and zero otherwise. The weights $v_t$ allow to emphasize regions of the window which are of special importance. In our experiments they are fixed to $v_t = 0.5 - 0.4|l-t|/l$. The hyperparameter $d$ determines the order to which local correlations are considered. The locality improved kernel can be considered as a special form of a *polynomial kernel*, where only a weighted subset of *monomers* is considered [1].

### *C. Markov Chain Model*

As a baseline classifier, we consider simple Markov models of the positive and negative sequences, see [29] for an introduction. We apply *inhomogeneous Markov chains*, also referred to as *weight array matrix*

*models*. Given a Markov chain $M$ of order $n$ over an alphabet $\mathscr{A}$ for strings of a fixed length $l$ (cf. [29, section 4.4.2] and [33]), the likelihood of a sequence $\mathbf{s}$ is given by

$$P^M(\mathbf{s}) = P_1^M(s_1) \cdot P_2^M(s_2 \,|\, s_1) \cdot \cdots \cdot P_n^M(s_n \,|\, s_1, \ldots, s_{n-1}) \cdot \prod_{i=n+1}^{l} P_i^M(s_i \,|\, s_{i-n}, \ldots, s_{i-1}) \ . \tag{16}$$

The conditional probabilities $P_i^M$ are the $\frac{|\mathscr{A}|^{n+1}-|\mathscr{A}|}{|\mathscr{A}|-1} + (l-n)|\mathscr{A}|^{n+1}$ parameters of the model and are estimated from the frequencies in the training data plus a *pseudo count* $c_{\text{pseudo}}$ (cf. [29, section 4.3.1]). For example, for a model of order $n = 2$ over the alphabet $\mathscr{A} = \{\mathsf{A}, \mathsf{T}, \mathsf{C}, \mathsf{G}\}$ and for $i > 2$ we have

$$P_i^M(s_i \,|\, s_{i-2}, s_{i-1}) = \frac{c_i^M(s_{i-2}\, s_{i-1}\, s_i) + c_{\text{pseudo}}}{c_i^M(s_{i-2}\, s_{i-1}\, \mathsf{A}) + c_i^M(s_{i-2}\, s_{i-1}\, \mathsf{C}) + c_i^M(s_{i-2}\, s_{i-1}\, \mathsf{G}) + c_i^M(s_{i-2}\, s_{i-1}\, \mathsf{T}) + 4c_{\text{pseudo}}} \ , \tag{17}$$

where $c_i^M(s_{i-2}\, s_{i-1}\, s_i)$ denotes the frequency of the subsequence $s_{i-2}\, s_{i-1}\, s_i$ at positions $i-2$ to $i$ in the data set used for building the model.

Let $M^+$ and $M^-$ be the Markov chain models built from the positive and negative examples in the training data, respectively. A sequence $\mathbf{s}$ is classified based on the sign of $\ln P^{M^+}(\mathbf{s}) - \ln P^{M^-}(\mathbf{s})$.

Our simple Markov chain model has only two hyperparameters, its order $n$ and the value of the pseudo count $c_{\text{pseudo}}$. The latter serves as a regularization parameter.

More sophisticated Markov models, for example *interpolated Markov models* or *interpolated context models* [39], [36], as well as hybrid methods combining Markov models with other machine learning techniques [33] are likely to increase the performance. However, similar motif extraction (see section III-B) and hybridization techniques would also improve the performance of the oligo kernel classifier—and benefit from accurate model selection. In this study, the experiments are restricted to the classifiers in their generic form, not only to make the improvements by our model selection approach directly accessible, but also because in our application the training data are sparse and thus too complex models derived from properties of the training data are prone to overfitting.

### *D. Model Selection*

We first describe our new model selection approach applied to the combined oligo kernel for TIS prediction. Then we describe how the model selection is done for the alternative models we consider for comparison, namely SVMs using oligo kernels with only few adapted parameters, SVMs using the locality improved kernel, and Markov chain models. In the experiments all model selection processes are repeated independently for the 50 training data sets.

*1) Oligo kernels:* We consider the combined oligo kernel (12) with $\kappa = 6$. The six smoothing parameters of $\tilde{k}_{\text{6-combined}}$ are adapted by gradient ascent on the kernel-target alignment. The regularization parameter $C$ of the SVM cannot be optimized using the kernel-target alignment, which is independent of the actual learning algorithm applied in the feature space. Therefore, the hyperparameter adaptation comprises two steps:

1) gradient-based optimization of the kernel parameters by maximizing the kernel-target alignment, and
2) adaptation of the regularization parameter $C$ by minimizing the classification error estimated by cross-validation using grid-search.

First, the kernel parameters (here 6) are adapted by optimizing the kernel-target alignment using 60 iterations of iRprop$^+$, a gradient-based algorithm [40]. All training examples are used to compute the kernel-target alignment. The $\sigma_i$ are initially set to 1. Second, the regularization parameter $C$ of the SVM is optimized using grid-search. We look at the grid-points $\{0.1 \cdot i \,|\, 1 \leq i \leq 50\}$. As a performance measure, we compute the mean classification error on the hold out data sets in a 5-fold cross-validation procedure (i.e., the training data set is split in 5 partitionings with pairwise disjoint hold out data sets of size $\ell/5$). Finally the SVM is trained using the adapted parameter set using the complete training data. The resulting classifier is evaluated on the previously unseen test data.

This general procedure performs grid-search in only one dimension (i.e., a line-search). The adaptation of the kernel parameters is completely decoupled from the grid search and does not require SVM training at all. We like to stress that the proposed method scales well with the number of kernel parameters. In contrast to methods that are solely based on grid-search, a large number of hyperparameters can be adapted.

For comparison, we also test the trimer oligo kernel $\tilde{k}_3$ as defined in (11). The model selection is done as for the combined kernel. The hyperparameter $\sigma_3$ is adjusted by gradient ascent on the kernel-target alignment and $C$ by grid-search as described above.

In order to get some insights about the objective function surface of the kernel-target alignment maximization problem and the robustness of our model selection approach, we conduct some additional experiments. We vary the single hyperparameter $\sigma_3$ of the trimer oligo kernel on a log scale and compute the corresponding kernel-target alignment on the first training data partition. Further, we consider 50 independent kernel-target alignment optimizations of the combined oligo kernel on this single data set

TABLE I

OPTIMIZED REGULARIZATION PARAMETER $C$ AND SMOOTHING PARAMETERS FOR THE COMBINED OLIGO KERNEL. ALL RESULTS REFER TO 50 TRIALS WITH DIFFERENT PARTITIONINGS OF THE DATA.

| | $C$ | $\sigma_1$ | $\sigma_2$ | $\sigma_3$ | $\sigma_4$ | $\sigma_5$ | $\sigma_6$ |
|---|---|---|---|---|---|---|---|
| mean | 0.998 | 0.015 | 0.073 | 1.748 | 2.134 | 2.323 | 2.334 |
| 25% quantile | 0.7 | 0.015 | 0.006 | 1.675 | 2.054 | 2.197 | 2.165 |
| median | 0.9 | 0.015 | 0.015 | 1.741 | 2.116 | 2.320 | 2.295 |
| 75% quantile | 1.2 | 0.015 | 0.015 | 1.811 | 2.179 | 2.418 | 2.450 |

starting from different initializations. The initial values for the $\sigma_i$ are drawn independently from a log-normal distribution, where the normal distribution has zero mean and standard deviation two.

*2) Locality Improved Kernel:* For comparison, we build $C$-SVMs based on the locality improved kernel as described in section IV-B. This kernel compares two sequences locally within a small window of length $2l+1$ around a sequence position. A second parameter $d$ controls the order of local correlations within a window. The parameters $l$ and $d$ are integers. Thus, the family of (standard) locality improved kernels has no appropriate differentiable structure. Therefore, gradient-based optimization cannot be applied directly and the parameters have to be adjusted by a direct (zeroth order) search method.

We consider $C \in \{0.002 \cdot i \,|\, 1 \leq i \leq 10\}$ and $l, d \in \{i \,|\, 1 \leq i \leq 6\}$ (this is a reasonable range, see [28]). Two different model selection strategies are compared. First, we pick $C$, $l$, and $d$ based on three-dimensional grid-search and 5-fold cross-validation as described above. Second, we adopt the model selection approach used for the oligo kernels and pick $l, d \in \{i \,|\, 1 \leq i \leq 6\}$ based on two-dimensional grid-search using the kernel-target alignment as performance criterion. That is, no SVMs are built in the process of choosing $l$ and $d$. The regularization parameter $C \in \{0.002 \cdot i \,|\, 1 \leq i \leq 10\}$ is then adjusted using grid-search.

*3) Markov Chain Model:* The order $n$ and the value of the pseudo count $c_{\text{pseudo}}$ are optimized using grid-search over the values $c_{\text{pseudo}} \in \{0.2 \cdot i \,|\, 1 \leq i \leq 10\}$ and $n \in \{i \,|\, 0 \leq i \leq 5\}$. Selection criterion is 5-fold cross-validation as described above.

### *E. Results & Discussion*

The optimized hyperparameters of the combined oligo kernel are shown in Table I. There is not much variability among the 50 trials with different data partitionings, that is, the model selection process was robust. In Fig. 3 we visualize the dependence between the kernel-target alignment of the trimer oligo

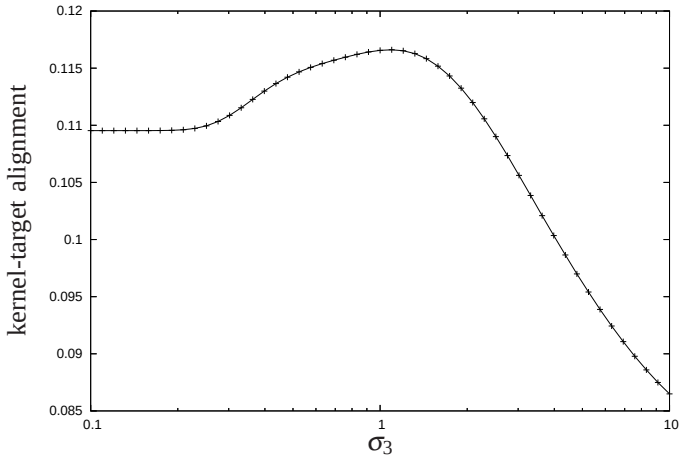


Fig. 3. Visualization of the search space when maximizing the kernel-target alignment for the trimer oligo kernel. The single kernel parameter $\sigma_3$ is varied on a log scale and the corresponding kernel-target alignment computed on the first training data partition is plotted.

TABLE II

THE STATISTICS OF THE FINAL SMOOTHING PARAMETERS AFTER MAXIMIZATION OF THE KERNEL TARGET ALIGNMENT BETWEEN THE COMBINED OLIGO KERNEL AND THE EMPIRICAL KERNEL OF THE FIRST TRAINING DATA PARTITION. THE VALUES REFER TO 50 TRIALS STARTING FROM DIFFERENT RANDOM INITIALIZATIONS.

| | $\sigma_1$ | $\sigma_2$ | $\sigma_3$ | $\sigma_4$ | $\sigma_5$ | $\sigma_6$ |
|---|---|---|---|---|---|---|
| mean | 0.010 | 0.008 | 1.662 | 2.086 | 2.267 | 2.136 |
| 25% quantile | 0.007 | 0.006 | 1.662 | 2.086 | 2.267 | 2.136 |
| median | 0.010 | 0.008 | 1.662 | 2.086 | 2.267 | 2.136 |
| 75% quantile | 0.014 | 0.012 | 1.662 | 2.086 | 2.267 | 2.136 |

kernel and the single smoothing parameter $\sigma_3$. In this concrete example, the objective function of the model selection process is unimodal. The same seems to be true for the combined oligo kernel, as evident from the results of repeated optimization considering a single training data set, but starting from random initial points. In all 50 trials the optimization ends up in approximately the same optimum, see Table II. The variance in the first two parameters is due to numerics, as the objective function becomes extremely flat for small values of $\sigma_i$, see Fig. 3. At least for the problem at hand, these results indicate that the objective function surfaces are not very rugged and that gradient-based algorithms seem to be appropriate for the optimization of the kernel-target alignment.

The final values for the smoothing parameters in Table I show that the positional uncertainty increases with the oligomer length. On the level of individual bases and dimers, the optimized kernels use Gaussians that are narrow peaks and virtually just count exact matches. However, there is a considerable increase

in $\sigma_K$ for $K \geq 3$. On the level of trimers and longer fragments, matching subsequences shifted by a few bases nucleotides contribute to the similarity of two sequences. Note that a $\sigma_i$-value of 2.5 implies that a subsequence shifted by three nucleotides has still $\approx 70\,\%$ of the contribution of an exact match in the kernel function (10).

TABLE III

THE RESULTS FOR THE FINAL HYPERPARAMETER CONFIGURATIONS OVER THE 50 PARTITIONS FOR THE TRIMER OLIGO KERNEL, THE LOCALITY IMPROVED KERNEL, AND THE MARKOV CHAIN MODEL.

| | 3mer oligo | | locality improved | | | Markov chain model | |
|---|---|---|---|---|---|---|---|
| | $C$ | $\sigma_3$ | $C$ | $l$ | $d$ | $n$ | $c_{\text{pseudo}}$ |
| mean | 0.786 | 1.197 | 0.00684 | 2 | 4.92 | 1.34 | 0.712 |
| 25% quantile | 0.5 | 1.128 | 0.004 | 2 | 5 | 1 | 0.2 |
| median | 0.7 | 1.198 | 0.006 | 2 | 5 | 1 | 0.6 |
| 75% quantile | 1.1 | 1.276 | 0.008 | 2 | 5 | 2 | 1.0 |

Table III shows the statistics of the final hyperparameters for the trimer oligo kernel, the locality improved kernel, and the Markov chain model. Again, there is only little variance. The order of the Markov chains is between one and two. One reason for the low order is of course the limited training data that does not allow for estimation of too many model parameters.

The classification performances of the different methods are shown in Table IV. The tables gives the mean values as well as 25%, 50% and 75% quantiles over the 50 partitions of the classification error on the test set (accuracy), specificity, sensitivity, and Matthews correlation coefficient [41]. Specificity is defined by $\mathrm{TN}/(\mathrm{TN}+\mathrm{FP})$, sensitivity by $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, and Matthews correlation coefficient by

$$\frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}\ , \tag{18}$$

where TP, TN, FP, and FN denote the true positives, true negatives, false positives, and false negatives, respectively.

In Fig. 4 the *ROC* (receiver operating characteristic) curves of the classifiers are shown. For the SVMs, the curves were obtained by simply varying the threshold parameter $b$ (see [42] for a more advanced approach). For the Markov chain model, a threshold $b$ parameter was introduced and adjusted, that is, a sequence was classified based on the sign of $\ln P^{M^+}(\mathbf{s}) - \ln P^{M^-}(\mathbf{s}) + b$. Each curve in Fig. 4 corresponds to the median of the 50 trials (similar to the attainment surfaces described in [43]).

TABLE IV

EXPERIMENTAL RESULTS IN TERMS OF CLASSIFICATION ACCURACY, SPECIFICITY, SENSITIVITY, AND MATTHEWS CORRELATION COEFFICIENT. THE MEAN VALUES AS WELL AS 25%, 50% AND 75% QUANTILES OVER 50 RUNS ARE LISTED. THE ACCURACY OF THE COMBINED OLIGO KERNEL IS STATISTICALLY SIGNIFICANTLY BETTER THAN THE ACCURACY OF THE OTHER METHODS (PAIRED WILCOXON RANK SUM TEST, $p < 0.001$).

| model | accuracy | specificity | sensitivity | correlation |
|---|---|---|---|---|
| SVM, trimer oligo kernel | 92.86 % | 95.97 % | 87.54 % | 83.84 % |
| 25% quantile | 92.48 % | 95.29 % | 85.89 % | 82.76 % |
| median | 92.84 % | 96.07 % | 87.84 % | 83.71 % |
| 75% quantile | 93.36 % | 96.55 % | 88.89 % | 85.00 % |
| SVM, combined oligo kernel | 93.30 % | 95.83 % | 88.96 % | 85.02 % |
| 25% quantile | 92.91 % | 95.41 % | 88.29 % | 84.13 % |
| median | 93.31 % | 95.80 % | 89.19 % | 85.00 % |
| 75% quantile | 93.71 % | 96.33 % | 90.09 % | 86.37 % |
| SVM, locality improved kernel | 92.54 % | 95.15 % | 88.10 % | 83.47 % |
| 25% quantile | 92.02 % | 94.55 % | 87.09 % | 82.50 % |
| median | 92.53 % | 95.03 % | 88.14 % | 83.46 % |
| 75% quantile | 93.03 % | 95.85 % | 89.19 % | 84.61 % |
| Markov chain model | 91.51 % | 92.01 % | 90.64 % | 82.69 % |
| 25% quantile | 90.86 % | 90.96 % | 89.49 % | 81.45 % |
| median | 91.42 % | 91.88 % | 90.69 % | 82.91 % |
| 75% quantile | 91.94 % | 93.01 % | 91.89 % | 83.68 % |

Through maximization of the kernel-target alignment the performance of the oligo kernel considerably improved. The accuracy of the optimized combined oligo kernel is statistically significantly better than the accuracy of all other methods in our study (paired Wilcoxon rank sum test, $p < 0.001$). The superior performance is also supported by the ROC curves in Fig. 4. The combined oligo kernel is clearly better than the trimer oligo kernel. This shows the benefits of considering more complex and flexible kernels in combination with an appropriate model selection strategy.

We considered two model selection methods for the SVMs with locality improved kernel, one based solely on the cross-validation classification error and one using the kernel-target alignment for adjusting the kernel parameters. In our application example, it turned out that the second method gave slightly better results although the same hyperparameter combinations were tested. This shows that model selection using the kernel-target alignment can lead to competitive results compared to cross-validation while being

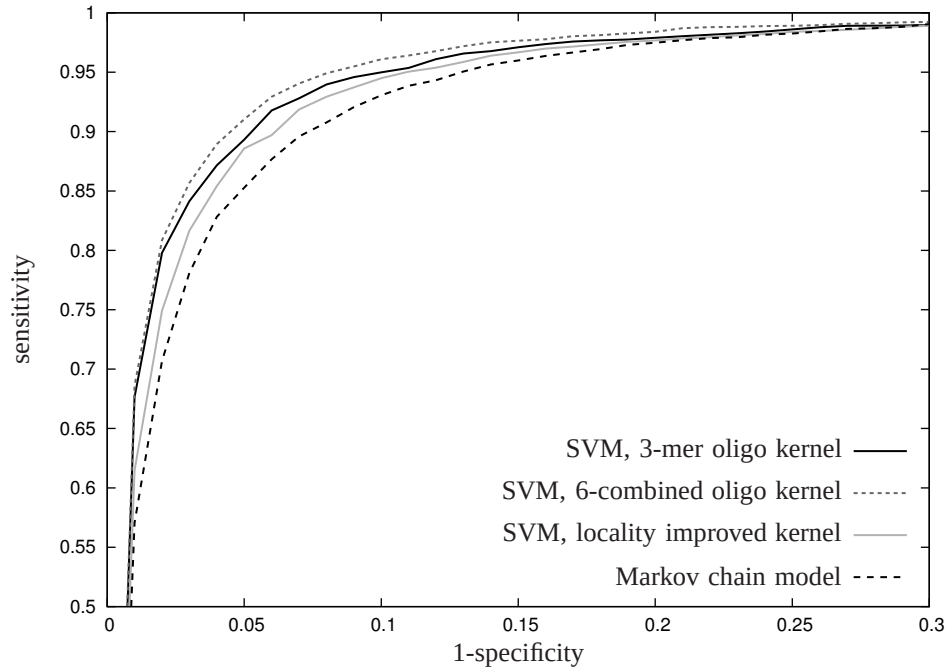


Fig. 4. Median ROC curves of the adapted classifiers based on 50 trials.

computationally less demanding. The three-dimensional grid-search involved testing 360 hyperparameter combinations and the cross-validation procedure required training of five SVMs per combination. In contrast, the kernel-target alignment was computed for 36 kernels and the subsequent adaptation of $C$ required building five SVMs only for each of the 10 possible values for the regularization parameter. In Table IV and Fig. 4 only the better results achieved by kernel-target alignment optimization are presented. Nonetheless, the locality improved kernel is worse than both the optimized trimer and the combined oligo kernel.

The inhomogeneous Markov chain models (weight array matrix models) serve as a baseline for the evaluation of the performance of the kernel classifiers. When adjusting the parameters of the Markov chain model properly, as done in this study, already very good results can be achieved. Still, the accuracy is significantly worse compared to all the kernel methods in our study (paired Wilcoxon rank sum test, $p < 0.001$). When looking at the results in Table IV the Markov chain models seem to perform well in terms of sensitivity, but the ROC curves in Fig. 4 reveal that for the same level of sensitivity the other classifiers show better specificity.

The kernel methods, in particular the combined oligo kernel with a mean accuracy of 93.30 %, give good classification results although the available training data set is rather small. The reasons might be that SVMs in general are a reasonable choice when dealing with small amounts of training data as well as

the appropriate model selection. Our parameterization of the oligo kernel provides the required flexibility. Of course, too much flexibility bears the risk of overfitting. And indeed, model selection as described in this study applied to a family of oligo kernels with over 60 parameters proposed in [35] overfits to limited training data.

## V. Conclusion

Biological sequence analysis using kernel methods benefits from a task-specific choice of the kernel function. We proposed gradient-based maximization of the kernel-target alignment for model selection. If the considered kernel space has a differentiable structure, this method can be applied to efficiently optimize multiple parameters. The kernel-target alignment can be maximized independently of the actual learning machine, in particular, solving quadratic optimization problems in each iteration is not required. Having such an efficient method at hand allows for extending the family of kernel functions considered during model selection.

The benefits of this additional flexibility and the power of the proposed model selection algorithm were demonstrated by adapting complex sequence kernels, namely oligo kernels. As an application example, we considered the prediction of bacterial gene starts using support vector machines. The classification performance improved significantly when the kernels were parameterized appropriately and when these parameters were chosen in a task-specific way by maximizing the kernel-target alignment. Analyzing the optimized kernel parameters can provide insights about the problem at hand. In our example, the results showed clear differences between the optimal position dependencies of different oligomer lengths.

## References

[1] B. Schölkopf, K. Tsuda, and J.-P. Vert, Eds., *Kernel Methods in Computational Biology*, ser. Computational Molecular Biology. MIT Press, 2004.

[2] P. Meinicke, M. Tech, B. Morgenstern, and R. Merkl, "Oligo kernels for datamining on biological sequences: A case study on prokaryotic translation initiation sites," *BMC Bioinformatics*, vol. 5, p. 169, 2004.

[3] S. Degroeve, B. D. Beats, Y. V. de Peer, and P. Rouzé, "Feature subset selection for splice site prediction." *Bioinformatics*, vol. 18 Suppl 2, pp. 75–83, 2002.

[4] C. Leslie, E. Eskin, and W. S. Noble, "The spectrum kernel: A string kernel for SVM protein classification." in *Proceedings of the Pacific Symposium on Biocomputing*, R. B. Altman, A. K. Dunker, L. Hunter, H. Lauerdale, and T. E. Klein, Eds. World Scientific, 2002, pp. 564–575.

[5] F. Markowetz, L. Edler, and M. Vingron, "Support vector machines for protein fold class prediction." *Biometrical Journal*, vol. 45, no. 3, pp. 377–389, 2003.

[6] H. Li and T. Jiang, "A class of edit kernels for SVMs to predict translation initiation sites in eukaryotic mRNAs," *Journal of Computational Biology*, vol. 12, no. 6, pp. 702–718, 2005.

[7] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 131–159, 2002.

[8] K.-M. Chung, W.-C. Kao, C.-L. Sun, and C.-J. Lin, "Radius margin bounds for support vector machines with RBF kernel," *Neural Computation*, vol. 15, no. 11, pp. 2643–2681, 2003.

[9] C. Gold and P. Sollich, "Model selection for support vector machine classification," *Neurocomputing*, vol. 55, no. 1-2, pp. 221–249, 2003.

[10] S. S. Keerthi, "Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1225–1229, 2002.

[11] H. Xiong, M. N. S. Swamy, and M. O. Ahmad, "Optimizing the kernel in the empirical feature space," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 460–474, 2005.

[12] T. Glasmachers and C. Igel, "Gradient-based adaptation of general gaussian kernels," *Neural Computation*, vol. 17, no. 10, pp. 2099–2105, 2005.

[13] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola, "On kernel-target alignment," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2001.

[14] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[15] C. S. Ong, A. J. Smola, and R. C. Williamson, "Learning the kernel with hyperkernels," *Journal of Machine Learning Research*, vol. 6, pp. 1043–1071, 2005.

[16] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[17] C. O. Gualerzi and C. L. Pon, "Initiation of mRNA translation in procaryotes," *Biochemistry*, vol. 29, no. 25, pp. 5881–5889, 1990.

[18] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. MIT Press, 1999, ch. 12, pp. 185–208.

[19] R.-E. Fan, P.-H. Chen, and C.-J. Lin, "Working set selection using the second order information for training support vector machines," *Journal of Machine Learning Research*, vol. 6, pp. 1889–1918, 2005.

[20] T. Glasmachers and C. Igel, "Maximum-gain working set selection for support vector machines," *Journal of Machine Learning Research*, vol. 7, pp. 1437–1466, 2006.

[21] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Advances in Computational Mathematics*, vol. 13, no. 1, pp. 1–50, 2000.

[22] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[23] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[24] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, USA: Springer-Verlag, 1995.

[25] F. Friedrichs and C. Igel, "Evolutionary tuning of multiple SVM parameters," *Neurocomputing*, vol. 64, no. C, pp. 107–117, 2005.

[26] T. P. Runarsson and S. Sigurdsson, "Asynchronous parallel evolutionary model selection for support vector machines," *Neural Information Processing – Letters and Reviews*, vol. 3, no. 3, pp. 59–68, 2004.

[27] Y. Baram, "Learning by kernel polarization," *Neural Computation*, vol. 17, pp. 1264–1275, 2005.

[28] 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*, vol. 16, no. 9, pp. 799–807, 2000.

[29] A. Krogh, "An introduction to hidden Markov models for biological sequences," in *Computational Methods in Molecular Biology*, S. L. Salzberg, D. B. Searls, and S. Kasif, Eds. Elsevier, 1998, ch. 4, pp. 45–63.

[30] S. L. Salzberg, "A method for identifying splice sites and translational start sites in eukaryotic mRNA," *Computer Applications in the Biosciences*, vol. 13, pp. 365–376, 1997.

[31] A. G. Pedersen and H. Nielsen, "Neural network prediction of translation initiation sites in eukaryotes: Perspectives for est and genome analysis," in *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 1997, pp. 226–233.

[32] A. G. Hatzigeorgiou, "Translation initiation start prediction in human cDNAs with high accuracy," *Bioinformatics*, vol. 18, pp. 343–350, 2002.

[33] J. C. Rajapakse and L. S. Ho, "Markov encoding for detecting signals in genomic sequences," *ACM Transactions on Computational Biology and Bioinformatics*, vol. 2, no. 2, pp. 131–142, 2006.

[34] M. Tech and P. Meinicke, "An unsupervised classification scheme for improving predictions of prokaryotic TIS," *BMC Bioinformatics*, no. 121, p. 7, 2006.

[35] B. Mersch, T. Glasmachers, P. Meinicke, and C. Igel, "Evolutionary optimization of sequence kernels for detection of bacterial gene starts," in *International Conference on Artificial Neural Networks (ICANN 2006)*, ser. LNCS, Kollias *et al.*, Eds., no. 4132. Springer-Verlag, 2006, pp. 827–836.

[36] A. L. Delcher, D. Harmon, S. Kasif, O. White, and S. L. Salzberg, "Improved microbial gene identification with GLIMMER," *Nucleic Acids Research*, vol. 27, no. 23, pp. 4636–4641, 1999.

[37] K. E. Rudd, "Ecogene: a genome sequence database for *Escherichia coli* K-12," *Nucleic Acids Research*, vol. 28, pp. 60–64, 2000, http://bmb.med.miami.edu/EcoGene/EcoWeb/.

[38] F. R. Blattner, G. Plunkett, C. A. Bloch, N. T. Perna, V. Burland, M. Riley, J. Collado-Vides, J. D. Glasner, C. K. Rode, G. F. Mayhew, J. Gregor, N. W. Davis, H. A. Kirkpatrick, M. A. Goeden, D. J. Rose, B. Mau, and Y. Shao, "The complete genome sequence of Escherichia coli K-12," *Science*, vol. 277, pp. 1453–1462, 1997, ftp://ftp.ncbi.nlm.nih.gov/genbank/genomes/Bacteria/Escherichia_coli_K127.

[39] S. L. Salzberg, A. L. Delcher, S. Kasif, and O. White, "Microbial gene identification using interpolated markov models," *Nucleic Acids Research*, vol. 26, no. 2, pp. 544–548, 1998.

[40] C. Igel and M. Hüsken, "Empirical evaluation of the improved Rprop learning algorithm," *Neurocomputing*, vol. 50, no. C, pp. 105–123, 2003.

[41] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta (BBA) – Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.

[42] T. Suttorp and C. Igel, "Multi-objective optimization of support vector machines," in *Multi-objective Machine Learning*, Y. Jin, Ed. Springer-Verlag, 2006, pp. 199–220.

[43] C. M. Fonseca and P. J. Fleming, "On the performance assessment and comparison of stochastic multiobjective optimizers," *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature (PPSN IV)*, pp. 584–593, 1996.

# APPENDIX

## A. Oligo Kernels Are Positive Definite Kernels

We show that the various oligo kernels are indeed positive definite kernel functions. We first consider some mathematical properties of oligo functions $\mu_\omega$ and the feature map $\Phi^K$. Because oligo functions are finite sums of Gaussians, they are infinite differentiable and square integrable, that is, $\int \mu_\omega^2(t)\mathrm{d}t < \infty$. This implies that oligo functions are elements of the Hilbert space $L_2$ with standard dot product $\langle f,g\rangle = \int_{\mathbb{R}} f(t)g(t)\mathrm{d}t$ for $f,g \in L_2$. Thus, $\Phi^K$ maps the sequences to the Hilbert space $(L_2)^m$ endowed with canonical dot product $\langle f,g\rangle = \int_{\mathbb{R}} f(t)\cdot g(t)\mathrm{d}t$ for $f,g \in (L_2)^m$, where $a\cdot b$ denotes the standard scalar product between $a,b \in \mathbb{R}^m$.

Computing $k_K(\mathbf{s}_i,\mathbf{s}_j)$ as defined in (10) corresponds to the dot product in the feature space $\mathscr{F} = L_2^m$ of the feature space representations of the sequences $\mathbf{s}_i,\mathbf{s}_j$. Therefore, $k_K$ is a positive definite kernel.

The normalized oligo kernel (11) is still a positive definite kernel, because in general it holds that if $k$ is a positive definite kernel on $\mathscr{X}$ then $k(x,y)/\sqrt{k(x,x)k(y,y)}$, $x,y \in \mathscr{X}$, is also a positive definite kernel. The combined oligo kernel (12) is positive definite, because in general it holds that if $k_1$ and $k_2$ are positive definite kernels on $\mathscr{X}$ then $k_{12}(x,y) = w_1k_1(x,y) + w_2k_2(x,y)$, $x,y \in \mathscr{X}, w_1,w_2 \in \mathbb{R}^+$, is also a positive definite kernel [23].

## B. Geometric View on Kernel-Target Alignment Maximization

Kernel-target alignment maximization aims at finding a kernel $k^*$ from a restricted family of "reasonable" kernel functions such that the Gram matrix induced by $k^*$ has the smallest distance to the empirical kernel matrix. The empirical kernel matrix $yy'$ is defined by the outer product of the class labels $y = (y_1,\ldots,y_\ell)'$ of the $\ell$ training patterns.

Formally, let $\mathscr{K}$ be the set of possible positive definite kernels on $\mathscr{X}$. Let $\mathscr{S} \subset \mathscr{K}$ be the parameterized family of kernel functions to which the model selection process is restricted. Given a set $D$ of training patterns, we define the equivalence relation $\sim_D$ on $\mathscr{K}$ by $k_1 \sim_D k_2$ if and only if $G_{k_1}/\sqrt{\langle G_{k_1},G_{k_1}\rangle} = G_{k_2}/\sqrt{\langle G_{k_2},G_{k_2}\rangle}$, where $G_{k_1}$ and $G_{k_2}$ are the Gram matrices on $D$ for the two kernels. We consider the quotient spaces $\mathscr{K}_D = \mathscr{K}/\sim_D$ and $\mathscr{S}_D = \mathscr{S}/\sim_D$. The distance between two equivalence classes

$[k_1],[k_2] \in \mathscr{K}_D$ with representatives $k_1$ and $k_2$ can be defined as

$$d([k_1],[k_2]) := \sqrt{\left\langle G_{k_1}/\sqrt{\langle G_{k_1},G_{k_1}\rangle} - G_{k_2}/\sqrt{\langle G_{k_2},G_{k_2}\rangle}, \langle G_{k_1}/\sqrt{\langle G_{k_1},G_{k_1}\rangle} - G_{k_2}/\sqrt{\langle G_{k_2},G_{k_2}\rangle}\right\rangle}$$
$$= \sqrt{2-2S(k_1,k_2)} \quad . \quad (19)$$

Thus, we seek a kernel $k^*$ minimizing $d([k],[\overline{yy}]) = \sqrt{2-2\hat{A}(k)}$, which is equivalent to maximizing $\hat{A}(k)$.

**Christian Igel** received his Diploma degree in computer science from the University of Dortmund, Germany, and his Doctoral degree from the Technical Faculty of the University of Bielefeld, Germany. He is junior professor for optimization of adaptive systems at the Institut für Neuroinformatik and faculty member of the International Graduate School of Neuroscience at the Ruhr-Universität Bochum. His research focuses on machine learning and information processing in biological systems. He is a senior member of the IEEE.

**Tobias Glasmachers** received his Diploma degree in mathematics from the Ruhr-Universität Bochum, Germany. He is now a PhD-Student at the Institut für Neuroinformatik in Bochum. He is interested in machine learning techniques, especially in kernel methods.

**Britta Mersch** received her Diploma degree in biomathematics from the University of Greifswald, Germany. She is a PhD-Student at the German Cancer Research Center in Heidelberg. Her research focuses on machine learning and bioinformatics, in particular the analysis of tissue and cancer specific gene expression.

**Nico Pfeifer** received his Master of Science degree in applied computer science from the University of Göttingen, Germany. He is a PhD-Student in the group of Oliver Kohlbacher at the Institut für Informatik in Tübingen. His main interests are in machine learning, computational proteomics and immunomics.

**Peter Meinicke** received the Diploma in informatics and his Doctoral degree from the University of Bielefeld. He is now a senior researcher in bioinformatics at the University of Goettingen. His research interests are focused on machine learning technology for the analysis of biological systems.