BMC Bioinformatics

RESEARCH Open Access

Exploiting physico-chemical properties in string kernels

Nora C Toussaint[1*], Christian Widmer[2], Oliver Kohlbacher[1], Gunnar Rätsch[2]

From Machine Learning in Computational Biology (MLCB) 2009
Whistler, Canada. 10-11 December 2009

Abstract

Background: String kernels are commonly used for the classification of biological sequences, nucleotide as well as amino acid sequences. Although string kernels are already very powerful, when it comes to amino acids they have a major short coming. They ignore an important piece of information when comparing amino acids: the physico-chemical properties such as size, hydrophobicity, or charge. This information is very valuable, especially when training data is less abundant. There have been only very few approaches so far that aim at combining these two ideas.

Results: We propose new string kernels that combine the benefits of physico-chemical descriptors for amino acids with the ones of string kernels. The benefits of the proposed kernels are assessed on two problems: MHC-peptide binding classification using position specific kernels and protein classification based on the substring spectrum of the sequences. Our experiments demonstrate that the incorporation of amino acid properties in string kernels yields improved performances compared to standard string kernels and to previously proposed non-substring kernels.

Conclusions: In summary, the proposed modifications, in particular the combination with the RBF substring kernel, consistently yield improvements without affecting the computational complexity. The proposed kernels therefore appear to be the kernels of choice for any protein sequence-based inference.

Availability: Data sets, code and additional information are available from http://www.fml.tuebingen.mpg.de/raetsch/suppl/aask. Implementations of the developed kernels are available as part of the Shogun toolbox.

Background

String kernels are a powerful and popular tool for machine learning in computational biology. They have been successfully applied to numerous applications ranging from protein remote homology detection [1-3], to gene identification [4-6], to sub-cellular location prediction [7,8] to drug design [9,10]. The different kernel formulations commonly exploit the sequential structure of the sequences and by doing so can effectively eliminate implausible features, leading to improved results. When using string kernels on protein sequences, one key disadvantage is that prior knowledge about the properties of individual amino acids (AAs), e.g., their size, hydrophobicity, secondary structure preference, cannot be easily incorporated. While these properties can be learned implicitly by the machine learning methods if the training data sets are large enough, it would be advantageous to include this information in the sequence representation. The goal of this work is to combine the benefits of string kernels with the ones of physico-chemical descriptors for AAs. The main idea is to replace the comparison of substrings, which is computed during kernel computation, with a term that takes the AA properties into account. While this seems quite simple at first sight, it is less so, when considering k-mers instead of single AAs. The key insight is how to compute the kernels such that the beneficial properties of sequence kernels do not get lost. In particular, we

* Correspondence: nora.toussaint@uni-tuebingen.de
[1]Center for Bioinformatics, Eberhard-Karls-Universität, Sand 14, 72076 Tübingen, Germany
Full list of author information is available at the end of the article

would like that either the use of uninformative descriptors (e.g., each AA corresponds to a unit vector) or the choice of distinct kernel parameters reduces the new kernel to the original string kernel.

String kernels for sequence classification

Kernels that have been proposed for classifying nucleic and amino acids can be divided into two main classes: (a) kernels describing the sequence content of sequences of varying length and (b) kernels for identifying localized signals within sequences of fixed length. The first class is typically used for classifying whole protein or mRNA sequences, while the second class is typically used to recognize a specific site in a window of fixed length sliding over a sequence.

Kernels describing ℓ-mer content

The so-called *spectrum kernel* was first proposed for classifying protein sequences [11]:

$$k_\ell^{\mathrm{sp}}(\mathrm{x},\mathrm{x}') = \left\langle \Phi_\ell^{\mathrm{sp}}(\mathrm{x}), \Phi_\ell^{\mathrm{sp}}(\mathrm{x}') \right\rangle, \tag{1}$$

where **x, x′** are two sequences over an alphabet Σ, e.g. protein or DNA sequences. Φ_ℓ^{sp} is a mapping of the sequence **x** into a $|\Sigma|^\ell$-dimensional feature-space. Each dimension corresponds to one of the $|\Sigma|^\ell$ possible strings s of length ℓ and is the count of the number of occurrences of s in **x.** It is

$$\begin{aligned}\Phi_\ell^{\mathrm{sp}}(\mathrm{x})_s &= \left|\#s \text{ in } \mathrm{x}\right| \\ &= \sum_{i=1}^{|\mathrm{x}|-\ell+1} \mathbf{I}(\mathrm{x}_{[i:i+\ell]} = s)\end{aligned} \tag{2}$$

where $\mathbf{x}_{[i:i+l]}$ is the substring of length ℓ of **x** at position i.

Several algorithms based on string or sparse data structures have been proposed to efficiently compute the above kernel and additional variants (for instance, with gaps, mismatches, mixed-order, etc.). The kernel in (1) can alternatively be written as

$$k_\ell^{\mathrm{spectrum}}(\mathrm{x},\mathrm{x}') = \sum_{s\in\Sigma^\ell} \left|\#s \text{ in } \mathrm{x}\right| \cdot \left|\#s \text{ in } \mathrm{x}'\right| \tag{3}$$

$$= \sum_{i=1}^{|\mathrm{x}|} \sum_{j=1}^{|\mathrm{x}'|} \mathbf{I}\left(\mathrm{x}_{[\mathrm{i:i}+\ell]} = \mathrm{x}'_{[\mathrm{j:j}+\ell]}\right). \tag{4}$$

Here, we consider all pairs of substrings at any position in each of the two input sequences. This formulation has the benefit that it makes the comparison between the substrings more explicit, which is needed in the derivation of the extensions.

Kernels for localized signals

The spectrum kernel is less well-suited for identifying localized signals in sequences, since the information about the position of the substring in the input sequences is ignored, i.e. lost, during kernel computation. Several kernels have been proposed to address this issue. Most notably the *weighted degree (WD) kernel* [12] and the *oligo kernel* [13]. Both kernels work on sequences of fixed length L and count co-occurring substrings in both sequences at the same or similar position. We will use the WD kernel as representative for localized signal kernels. It is defined as

$$k_\ell^{\mathrm{wd}}(\mathrm{x},\mathrm{x}') = \sum_{d=1}^{\ell} \beta_d \sum_{i=1}^{L-d+1} \mathbf{I}\left(\mathrm{x}_{[\mathrm{i:i+d}]} = \mathrm{x}'_{[\mathrm{j:j+d}]}\right) \tag{5}$$

where $\beta_d = 2\frac{\ell - d + 1}{\ell^2 + \ell}$ is the weighting of the substring lengths. The WD kernel is quite related to the spectrum kernel formulation in (4), where we consider only the ℓ-mers occurring at the same position, i.e., where $i = j$. The oligo kernel is similar in spirit but it also compares substrings at different positions.

Incorporation of knowledge on AA properties

In this work we propose modifications to existing string kernels that supplement the kernels' beneficial properties by incorporating prior knowledge on physico-chemical and other properties of AAs. Previous work on incorporating prior knowledge has been either focused on using physico-chemical properties for single AAs, i.e., ignoring the sequential nature of the sequences (e.g., [14,15]), or took advantage of Blast or PSI-Blast profiles for improving spectrum kernels [2,3,16]. We propose a complementary approach of employing physico-chemical or other information to refine the similarity between two substrings used in most existing string kernels. We illustrate the usefulness of these modifications for both classes of string kernels on two problems: (a) the prediction of MHC-binding peptides as an example for localized signals and (b) protein fold classification as an example for ℓ-mer content.

Methods

Idea

The string kernels described above (cf. (4),(5)) have in common that they compare substrings of length l between the two input sequences **x** and **x′**. The involved term $\mathbf{I}(\overline{\mathrm{x}} = \overline{\mathrm{x}}')$ can equivalently be written as:

$$\mathbf{I}(\overline{\mathrm{x}} = \overline{\mathrm{x}}') = \left\langle \Phi_\ell(\overline{\mathrm{x}}), \Phi_\ell(\overline{\mathrm{x}}') \right\rangle,$$

where $\overline{\mathrm{x}}, \overline{\mathrm{x}}' \in \Sigma^\ell$ and $\Phi_\ell(\overline{\mathrm{x}}) \in \mathbb{R}^{|\Sigma^\ell|}$.

$\Phi_\ell(\overline{\mathrm{x}})$ can be indexed by a substring $s \in \Sigma^\ell$ and is defined as $\Phi_\ell(\overline{\mathrm{x}})_s - 1$, if $s = \overline{\mathrm{x}}$ and 0 otherwise. For the sake of the derivation, let us consider $\Phi_1 : \Sigma \mapsto \{0,1\}$, generating a simple encoding of the letters into $|\Sigma|$-dimensional unit vectors. It can be easily seen that we can rewrite the substring comparison as

$$\mathbf{I}(\bar{\mathrm{x}} = \bar{\mathrm{x}}') = \prod_{l=1}^{\ell} \left\langle \Phi_1(\bar{\mathrm{x}}_l), \Phi_1(\bar{\mathrm{x}}'_l) \right\rangle,$$

The main problem of using Φ_1 as the basis of substring comparisons, is that it ignores the relations between the letters in the alphabet. While this is a negligible problem for nucleotide sequences where each nucleotide is unique, it is important to consider relatedness between AAs. The main idea of this work is to replace Φ_1 with a feature map Ψ that takes relations between the AAs into account. One way is to use physico-chemical descriptors of AAs, such as [17]. Alternatively, one may use AA substitution matrices for defining amino acid similarities, as e.g. done in [18]. The feature space is then not spanned by $|\Sigma|^{\ell}$ different combinations of letters, but by D^l, where D is the number of properties used to describe the AA. This leads to the following kernel on AA substrings:

$$k_{\ell}^{\Psi}(\bar{\mathrm{x}}, \bar{\mathrm{x}}') = \prod_{l=1}^{\ell} \left\langle \Psi(\bar{\mathrm{x}}_l), \Psi(\bar{\mathrm{x}}'_l) \right\rangle, \qquad (6)$$

Using the feature representation corresponding to this kernel, we can now recognize sequences of AAs that have certain properties (e.g. first AA: hydrophobic, second AA: large, third AA: positively charged, etc.): There is a feature induced in the kernel corresponding to all combinations of products of features involving exactly one AA property per substring position. For instance, when considering products of the form $(\mathbf{x}_{1,1} + \mathbf{x}_{1,2} +\ldots+ \mathbf{x}_{1,n}) \cdot (\mathbf{x}_{2,1} +\mathbf{x}_{2,2} +\ldots+ \mathbf{x}_{2,n}) \cdot (\mathbf{x}_{3,1} + \mathbf{x}_{3,2} +\ldots+ \mathbf{x}_{3,n})$, then we get n^3 different monomials which each use exactly one of the n features from the three different groups. There are no monomials $\mathbf{x}_{i,j}\mathbf{x}_{i,k}$ for any $i = 1,\ldots,3$ and j, $k = 1,\ldots,n$.

If one wants to additionally allow the combination of several properties from every position, then the following two formulations are suitable: The first is based on the polynomial kernel:

$$k_{\ell,d}^{\Psi}(\bar{\mathrm{x}}, \bar{\mathrm{x}}') = \left(\sum_{l=1}^{\ell} \left\langle \Psi(\bar{\mathrm{x}}_l), \Psi(\bar{\mathrm{x}}'_l) \right\rangle \right)^d, \qquad (7)$$

and the second on the RBF kernel:

$$k_{\ell,\sigma}^{\Psi}(\bar{\mathrm{x}}, \bar{\mathrm{x}}') = \exp\left(-\frac{\sum_{l=1}^{\ell} \left\| \Psi(\bar{\mathrm{x}}_l) - \Psi(\bar{\mathrm{x}}'_l) \right\|^2}{\sigma^2} \right). \qquad (8)$$

Both kernels induce a considerably richer feature space, which can be beneficial for accurate classification of sequences.

AA substring kernel for localized string kernels

Replacing the substring comparison $\mathbf{I}(\bar{\mathrm{x}} = \bar{\mathrm{x}}')$ with the more general formulation in (6), (7), or (8) together with an informed choice of features $\Psi(a)$ for each letter $a \in \Sigma$ (i.e. for each AA), directly implies a generalized form of the string kernels described above. For the WD kernel we can write:

$$k_{\ell}^{\mathrm{wd},\Psi}(\mathrm{x}, \mathrm{x}') = \sum_{d=1}^{\ell} \beta_d \sum_{i=1}^{L-d+1} k_d^{\Psi}(\mathrm{x}_{[i:i+d]}, \mathrm{x}'_{[i:i+d]}). \qquad (9)$$

$k_{\ell}^{\mathrm{wd},\Psi}$ is a linear combination of kernels and therefore a valid kernel [19]. Independent of the choice of AA substring kernel, the modified WD kernel can be computed efficiently, with a complexity comparable to that of the original.

Of particular interest is the *WD-RBF kernel*, i.e. the combination of the WD kernel and the RBF AA substring kernel:

$$k_{\ell,\sigma}^{\mathrm{wd},\Psi}(\mathrm{x}, \mathrm{x}') = \sum_{d=1}^{\ell} \beta_d \sum_{i=1}^{L-d+1} \exp\left(-\frac{\sum_{j=1}^{d} \left\| \Psi(\mathrm{x}_j) - \Psi(\mathrm{x}'_j) \right\|^2}{\sigma^2} \right). \qquad (10)$$

For $\sigma \to 0$ and an encoding Ψ with $\Psi(a) = \Psi(b)$ if and only if $a = b$, the WD-RBF kernel corresponds to the WD kernel: the RBF AA substring kernel will be one only if the substrings are identical, otherwise it will be zero.

Relation to non-substring-based kernels

When considering kernels for sequences of fixed length L, one may alternatively consider a representation of the sequence as vector of the physico-chemical properties of all sequence elements/AAs, i.e. $\bar{\Psi}(\mathrm{x}) = (\boldsymbol{\Psi}(\mathrm{x}_1), \ldots, \boldsymbol{\Psi}(\mathrm{x}_\mathrm{L}))$. Then one may use a standard kernel to compute similarities between the sequences, as, e.g., done in [20]. When using the polynomial kernel as basis, this would lead to the following kernel:

$$k_d^{\mathrm{poly},\Psi}(\mathrm{x}, \mathrm{x}') = \left\langle \bar{\Psi}(\mathrm{x}), \bar{\Psi}(\mathrm{x}') \right\rangle^d = \left(\sum_{l=1}^{L} \left\langle \Psi(\mathrm{x}_l), \Psi(\mathrm{x}'_l) \right\rangle \right)^d. \qquad (11)$$

For the RBF kernel we get analogously,

$$k_d^{\mathrm{RBF},\Psi}(\mathrm{x}, \mathrm{x}') = \exp\left(-\frac{\left\| \bar{\Psi}(\mathrm{x}) - \bar{\Psi}(\mathrm{x}') \right\|^2}{\sigma^2} \right) = \exp\left(-\frac{\sum_{l=1}^{L} \left\| \Psi(\mathrm{x}_l) - \Psi(\mathrm{x}'_l) \right\|^2}{\sigma^2} \right). \qquad (12)$$

Please note that here we use the full sequence and do not separately consider subsequences. Both kernels consider higher order correlations between properties of the sequence at arbitrary position in the sequence. Hence, the sequential nature of the sequences is not fully taken into account (particularly important for long sequences).

AA substring kernel for ℓ-mer content string kernels

The AA substring kernels (6), (7), (8) can be combined with the spectrum kernel (1), (4) analogously to the combination with the WD kernel. For instance in combination with the RBF substring kernel, we arrive at:

$$k_\ell^{\text{sp-rbf}}(\mathrm{x},\mathrm{x}') = \sum_{i=1}^{|\mathrm{x}|-\ell+1} \sum_{j=1}^{|\mathrm{x}'|-\ell+1} \exp\left(-\frac{\sum_{j=1}^{\ell} \left\| \Psi(\mathrm{x}_i) - \Psi(\mathrm{x}'_j) \right\|^2}{\sigma^2} \right). \tag{13}$$

As before, for $\sigma \to 0$, the above formulation is identical to the original spectrum kernel. A drawback of this approach is, however, that one now has to compute the substring comparisons for every pair of occurring substrings. Hence, the computational complexity, $O(|\mathbf{x}| \cdot |\mathbf{x}'|)$, is much higher than for the original spectrum kernel and makes this kernel impractical.

In order to reduce complexity we turn to modifications of the spectrum kernel: the *mismatch kernel* [21] and the *profile kernel* [2].

The mismatch kernel

While the spectrum kernel only considers pairs of identical ℓ-mers, the mismatch kernel allows for some degree of mismatching. Instead of counting occurences of ℓ-mers s in $\mathbf{x}$ it counts the occurences of ℓ-mers that differ from s by at most m mismatches. The mismatch kernel is defined as

$$k_{(\ell,m)}^{\text{mm}}(\mathrm{x},\mathrm{x}') = \left\langle \Phi_{(\ell,m)}(\mathrm{x}), \Phi_{(\ell,m)}(\mathrm{x}') \right\rangle, \tag{14}$$

and

$$\Phi_{(\ell,m)}(\mathrm{x})_s = \sum_{i=1}^{|\mathrm{x}|-\ell+1} \phi_s(\mathrm{x}_{[i:i+\ell]}) \tag{15}$$

with $\varnothing_s(\mathbf{x}_{[i:i+l]}) = 1$ if $\mathbf{x}_{[i:i+l]}$ belongs to the mismatch neighbourhood $N_{(l,m)}(s)$, i.e. differs from s in at most m positions. Otherwise, $\varnothing_s(\mathbf{x}_{[i:i+l]}) = 0$. Thus, we can write alternatively:

$$\Phi_{(\ell,m)}(\mathrm{x})_s = \sum_{i=1}^{|\mathrm{x}|-\ell+1} \mathbf{I}(\mathrm{x}_{[i:i+\ell]} \in N_{(\ell,m)}(s)). \tag{16}$$

Combination of an AA substring kernel with the (ℓ,m)-mismatch kernel limits comparisons to those ℓ-mer pairs with at most $2m$ mismatches as opposed to all ℓ-mer pairs for the spectrum kernel. Employing the mismatch tree data structure from [21], the generalized mismatch kernel can be calculated efficiently with a complexity of $O(|\Sigma|^m l^m\ (|\mathbf{x}| + |\mathbf{x}'|))$. The (ℓ, m)-mismatch tree is a tree representation of the feature space: each leaf represents a fixed ℓ-mer feature s. In order to benefit from this feature space-based data structure, it suggests itself to apply the generalization to the feature map $\Phi_{(l,m)}(\mathbf{x})_s$ (16). Plugging one of the AA substring kernels into (16) yields

$$\Phi_{(\ell,m)}^{\Psi}(\mathrm{x})_s = \sum_{i=1}^{|\mathrm{x}|-\ell-1} k_\ell^{\Psi}(\mathrm{x}_{[i:i+\ell]}, s) \cdot \mathbf{I}(\mathrm{x}_{[i:i+\ell]} \in N_{(\ell,m)}(s)). \tag{17}$$

Rather than simply counting similar substrings this feature representation accounts for the degree of similarity: similar substrings contribute stronger than dissimilar ones. This strategy is particularly beneficial, when allowing many mismatches.

Once again, the combination with the RBF AA substring kernel, namely the *mismatch-RBF kernel,* is of particular interest. The corresponding feature map is defined as

$$\Phi_{(\ell,m)}^{\Psi}(\mathrm{x})_s = \sum_{i=1}^{|\mathrm{x}|-\ell-1} \exp\left(-\frac{\sum_{j=1}^{\ell} \left\| \Psi(\mathrm{x}_j) - \Psi(s_j) \right\|^2}{\sigma^2} \right) \cdot \mathbf{I}(\mathrm{x}_{[i:i+\ell]} \in N_{(\ell,m)}(s))). \tag{18}$$

For $\sigma \to \infty$ it corresponds to the mismatch feature map (16) since the RBF AA substring kernel will be one for all substring pairs.

The profile kernel

Just like the spectrum and the mismatch kernel, the profile kernel [2] was proposed for protein classification and remote homology detection. The main difference between the mismatch and profile kernel is the definition of the neighbourhood. For the profile kernel one uses the *positional mutation neighbourhood* of $\mathbf{x}$ based on blast or PSI-blast profiles $p(\mathbf{x_i}, k)$ for each position i of $\mathbf{x}$ and for each letter $k \in \Sigma$:

$$P_{\ell,\delta}(\mathrm{x}_{[i:i+\ell]}) = \left\{ s \in \Sigma^\ell : -\sum_{j=1}^{\ell} \log p(\mathrm{x}_\mathrm{j}, \mathrm{s}_\mathrm{j}) < \delta \right\}, \tag{19}$$

where δ defines the "radius" of the mutation neighbourhood [2]. Then the feature map and kernel, respectively, are defined as

$$\Phi_{(\ell,\tau)}(P(\mathrm{x}))_s = \sum_{i=1}^{|\mathrm{x}|-\ell+1} \mathbf{I}(s \in P(\mathrm{x}_{[i:i+\ell]})) \tag{20}$$

and

$$k_{(\ell,\tau)}^{\text{profile}}(P(\mathrm{x}), P(\mathrm{x}')) = \left\langle \Phi_{(\ell,\tau)}(P(\mathrm{x})), \Phi_{(\ell,\tau)}(P(\mathrm{x}')) \right\rangle. \tag{21}$$

In order to incorporate AA properties, we propose to modify (20) analogously to the mismatch kernel:

$$\Phi_{(\ell,\tau)}(P(\mathrm{x}))_s = \sum_{i=1}^{|\mathrm{x}|-\ell+1} k_\ell^\Psi(\mathrm{x}_{[i:i+\ell]}, s) \cdot \mathbf{I}(s \in P(\mathrm{x}_{[i:i+\ell]}) \quad (22)$$

The second term determines whether the substring is within the mutation neighbourhood and should be counted and the first term determines the contribution of the substring based on AA similarities. This kernel can be computed as efficient as the original profile kernel. Since the elements in the neighbourhood are weighted based on AA property similarity, the kernel may be able to take advantage of larger neighbourhoods.

The profile kernel is similar to the profile-based direct kernels described in [16] and similar ideas to incorporate AA properties can be applied there as well. The profile and mismatch kernel have, however, the advantage that they allow for an efficient computation using the data structures proposed in [2,22]. These data structures unfortunately are not applicable to the profile kernel formulations in [16].

Experimental methods

Data

We evaluate the performance of the proposed kernels on two problems: the kernels for localized signals on MHC-peptide binding classification, and the kernels describing ℓ-mer content on protein classification. For MHC-peptide binding experiments we utilized the IEDB benchmark data set from Peters *et al.* [23]. It contains quantitative binding data (IC_{50} values) of nonameric peptides with respect to various MHC alleles. Peptides with IC_{50} values greater than 500 were considered non-binders, all others binders. Protein classification data was taken from the supplementary material of [3]. This commonly used data set comprises 7,329 protein domains from 54 families. Corresponding profile information was taken from [http://cbio.mskcc.org/leslielab/software/string-kernels].

Physico-chemical descriptors

A wide range of physico-chemical descriptors of AAs have been published. Many of them can be obtained from the amino acid index database (AAIndex) [24]. Within this work we use three sets of descriptors: (1) five descriptors derived from a principal component analysis of 237 physico-chemical properties taken from the AAIndex [17] (*pca*), (2) three descriptors representing hydrophobicity, size, and electronic properties (*zscale*), and (3) 20 descriptors corresponding to the respective entries of the Blosum50 substitution matrix [25] (*blosum50*).

Evaluation of string kernels for localized signals

Performance analysis. Preliminary experiments on three human MHC alleles (A*2301, B*5801, A*0201) were carried out to analyze the performance of the different kernels WD (5), RBF (12), poly (11), WD-RBF (10), WD-poly (as WD-RBF, but with polynomial substring kernel) combined with different encodings (*pca, zscale, blosum50*). The alleles were chosen to comprise a small data set (A*2301, 104 examples) as well as a medium (B*5801, 988 examples) and a large (A*0201, 3,089 examples) data set from the IEDB benchmark [23]. Performances of the WD kernel and the WD-RBF kernel with *blosum50* encoding were subsequently analyzed on all 35 human MHC alleles contained in the IEDB benchmark. We used two times nested 5-fold cross-validation, i.e. two nested cross-validation loops, to (1) perform model-selection over the kernel and regularization parameters (inner loop), (2) estimate the prediction performance (outer loop) (see, e.g., page S30 of the supporting online material of [26]). Performance is measured by averaging the area under the ROC curve (auROC).

Learning curve analysis. The performance dependence on the amount of training data was analyzed on allele A*0201 in 100 runs of two times nested 5-fold cross-validation to average over different data splits to reduce random fluctuations of the performance values. Performance is measured by averaging the area under the ROC curve (auROC). In each run, thirty percent of the available data was used for testing. From the remaining data training sets of different sizes (20, 31, 50, 80, 128, 204, 324, 516, 822, 1,308) were selected randomly.

Evaluation of string kernels describing l-mer content

Mismatch kernel. For the comparison of the mismatch kernel and the mismatch-RBF kernel, protein classification data and experimental setup were taken from the supplementary material of [3]. The ROC_{50} score, i.e. the area under the ROC curve computed up to the first 50 false positives, is used as performance measure.

Profile kernel. For the comparison of the profile kernel and the profile-RBF kernel, protein classification data and experimental setup were taken from the supplementary material of [3]. Corresponding PSI-blast profiles were taken from [27]. The ROC_{50} score is used as performance measure.

SVM computations

All SVM computations were performed using the Matlab interface of the freely available large scale machine learning toolbox Shogun [28]. All used kernels are implemented as part of the toolbox.

Results and discussion

The main goal of this work is the methodological improvement of existing string kernels by incorporation of prior knowledge on AA properties. In order to analyze the benefits of the proposed modifications we conducted performance comparisons between the original and the modified string kernels.

String kernels for localized signals

The prediction of MHC-binding peptides is one of the major problems in computational immunology, highly relevant for rational vaccine design. MHC-I molecules bind small peptides derived from intracellular proteins and present them on the cell surface for surveillance by the immune system. Given a set of peptide sequences one would like to predict whether they bind to a certain MHC-I molecule. Since the majority of binders are of length nine, the application of kernels for localized signals suggests itself. For a preliminary analysis we chose three human MHC alleles from the IEDB benchmark data set: A*2301 (104 peptides), B*5801 (988 peptides), and A*0201 (3,089 peptides). The performance of various kernels utilizing sequential structure only (WD kernel), AA properties only (RBF, poly), and a combination of both (WD-RBF, WD-poly) was validated on these alleles. We used three different encodings of AA properties. Cross-validation results are given in Table 1. Best performance is achieved by a different kernel type for each of the alleles: poly (*pca*) for A*2301, RBF (*blosum50*) for B*5801 and WD-RBF (*blosum50*) for A*0201. The latter performs second-best on A*2301 and B*5801. As for the benefits of the modification of the WD kernel, the WD-poly and WD-RBF kernels outperform the WD kernel in 17 out of 18 cases. From Table 1 we can observe the trend that the kernels that use AA properties benefit more for smaller datasets. To validate this hypothesis, we performed a learning curve analysis for WD and WD-RBF (*blosum50*) on A*0201, the allele with the highest number of peptides in the IEDB benchmark data set. Figure 1 shows the mean auROCs with confidence intervals $(\sigma / \sqrt{n})$ over 100 cross-validation runs. We can clearly observe that the fewer examples are available for learning, the stronger is the improvement of the WD-RBF kernel over the WD kernel. Intuitively this makes sense, as the more data is available, the easier it will be to infer the relation of the AAs from the sequences in the training data alone.

Table 1 Performances of kernels utilizing sequential structure and/or AA properties on three MHC alleles

KERNEL	A*2301		B*5801		A*0201	
	auROC	(std)	auROC	(std)	auROC	(std)
WD	0.7307	(0.0900)	0.9314	(0.0279)	0.9485	(0.0076)
Poly (*pca*)	**0.8363**	(0.0808)	0.9428	(0.0336)	0.9354	(0.0111)
Poly (*zscale*)	0.7964	(0.0727)	0.8778	(0.0637)	0.9052	(0.0070)
Poly (*blosum50*)	0.8220	(0.0442)	0.4948	(0.0560)	0.4729	(0.0246)
RBF (*pca*)	0.8277	(0.0904)	0.9396	(0.0303)	0.9345	(0.0114)
RBF (*zscale*)	0.7847	(0.0787)	0.9235	(0.0347)	0.9157	(0.0072)
RBF (*blosum50*)	0.8204	(0.0864)	0.9509	(0.0317)	**0.9520**	(0.0072)
WD-Poly (*pca*)	0.7879*	(0.0858)	0.9406*	(0.0319)	0.9495*	(0.0084)
WD-Poly (*zscale*)	0.7983*	(0.0902)	0.9499*	(0.0348)	0.9483	(0.0073)
WD-Poly (*blosum50*)	0.8307*	(0.1077)	0.9491*	(0.0224)	0.9490*	(0.0070)
WD-RBF (*pca*)	0.8133*	(0.0806)	0.9510*	(0.0265)	0.9486*	(0.0051)
WD-RBF (*zscale*)	0.7782*	(0.1222)	0.9487*	(0.0434)	0.9500*	(0.0074)
WD-RBF (*blosum50*)	0.8312*	(0.0993)	**0.9571***	(0.0265)	0.9503*	(0.0067)

auROCs and standard deviation were determined in two times nested 5-fold cross-validation. Best (bold) and second-best (underlined) performances per MHC allele are highlighted. An asterisk marks performance improvement due to the proposed modifications.

The preliminary analysis showed the WD-RBF kernel with *blosum50* encoding to perform best. For a more comprehensive comparison, performance of WD and WD-RBF (*blosum50*) kernels were assessed on all 35 human MHC alleles from the IEDB benchmark. For 24 alleles WD-RBF outperforms WD (Fig. 2). This is significant with respect to the binomial distribution (p-Value = 0.0083).

Finally, we compare our results with the ones obtained using a multi-task learning (MTL) method for MHC classification described in [9]. Here, the authors used two kernels, one to define the similarity between examples and one to define the similarity between tasks. They report an auROC of 90.3% using two string kernels. When using the WD-RBF for computing the similarity between the examples, we can slightly improve upon their performance to 90.5% (data splits and model selection as in [9]). Hence, the AA property-enhanced kernels once more have a slight, but consistent advantage over the base-line kernels. Besides the performance improvement, the modified WD kernel allows, at least theoretically, for the extraction of biological insights: employing an analysis method analogous to [29] individual patterns of AA properties that are relevant for the classification can be extracted.

String kernels describing ℓ-mer content

To show that also the modification of kernels for describing ℓ-mer content of sequences has desirable properties, we chose the problem of protein remote homology detection. Here, the task is to classify proteins into folds, super-families or families based on their sequence. This problem has been previously tackled in a series of papers in [11,21,22] which suggested the spectrum kernel, followed by the mismatch kernel and finally the profile kernel. The profile kernel already uses AA similarities based on blast or PSI-blast profiles which lead to significant improvements. Here, we would like to illustrate that using the AA property-enhanced versions of these kernels can still lead to an improvement. We chose the family classification task for this

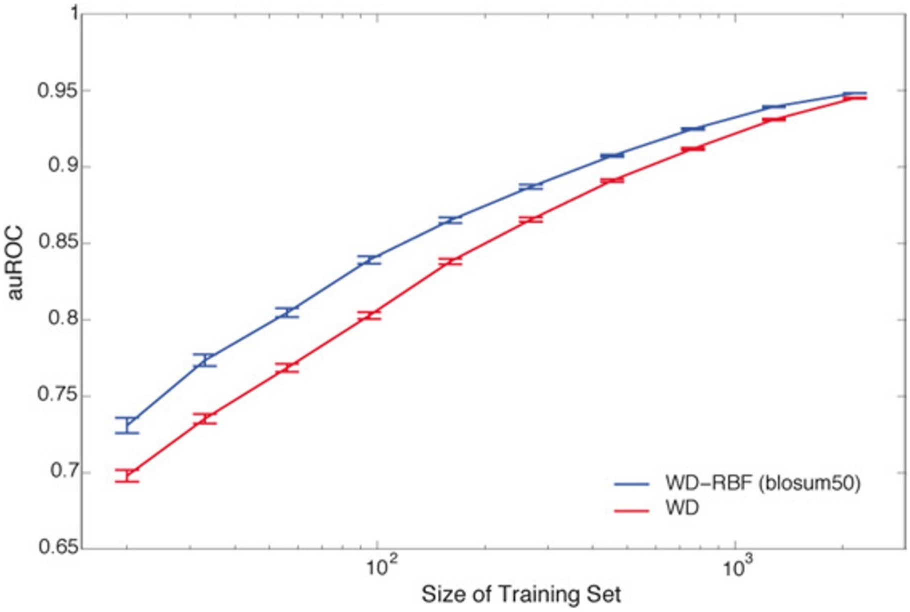

Figure 1 Learning Curve Analysis on MHC allele A*0201. Shown are areas under the ROC curves averaged over 100 different test splits (30%) and for increasing numbers of training examples (up to 70%). The training part was used for training and model selection using 5-fold cross-validation.

analysis since it was considered in all mentioned previous studies.

Table 2 shows the average auROC$_{50}$ score over the 54 families we obtained for the family classification problem. Furthermore, the number of times for which each method outperforms its counterpart is displayed. We compare the *spectrum kernel* [11] with the *spectrum-RBF kernel* as in (13) with *pca* features, the *mismatch kernel* [22] with the *mismatch-RBF kernel* as in (18); and the *profile kernel* [21] with the *profile-RBF kernel* as in (22). For all three cases we find that the AA property-enhanced kernels improve the original kernels. For *spectrum* and *mismatch* kernel these improvements are significant with respect to the binomial distribution. Most notably, the performance of the spectrum kernel can be drastically improved from 15.1% to 42.1%. However, the more sophisticated the original kernel already is, the smaller is the improvement that can be achieved by using additional AA property information.

In summary, in our experiments we can observe that the newly proposed kernels lead to consistently better

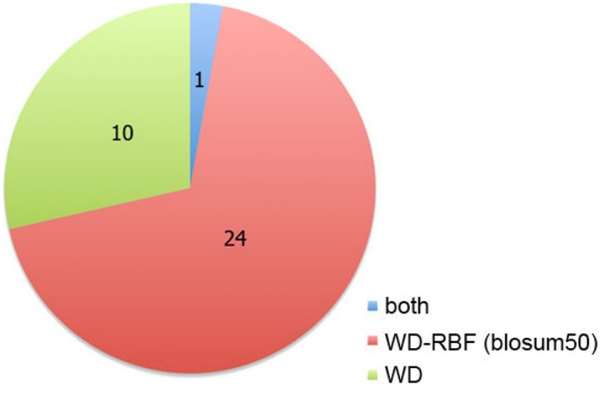

Figure 2 Performance of WD and WD-RBF (*blosum50*) kernels on human MHC alleles from the IEDB benchmark data set. The pie chart displays the number of alleles for which the WD (green) and the WD-RBF (red) performed best, respectively, and the number of alleles for which they performed equally (blue).

Table 2 Comparison of kernels for ℓ-mer content with their AA-property enhanced counterparts

Method	auROC$_{50}$	#Wins
Spectrum (ℓ = 5)	15.2%	7/54
Spectrum-RBF (ℓ = 5, σ = 1)	42.1%	45/54
Mismatch (ℓ = 5, m = 1)	42.3%	13/54
Mismatch-RBF (ℓ = 5,m = 1, σ = 1)	43.6%	36/54
Profile (ℓ = 5, τ = 7.5)	82.1%	3/54
Profile-RBF (ℓ = 5,τ = 7.5, σ = 100)	82.2%	10/54

Comparison of the three kernels proposed in [11,21,22], with their AA-property enhanced counterparts for remote homology detection of 54 protein families. auROC$_{50}$ is the average auROC$_{50}$ score and #Wins the number of families for which each method outperforms its counterpart (Spectrum vs. Spectrum-RBF, Mismatch vs. Mismatch-RBF, Profile vs. Profile-RBF). The kernels taking advantage of AA properties lead to a higher average accuracy in all three cases (p-Values: 6.92 10^{-8} for *spectrum*, 0.0045 for *mismatch*, and 1.0 for *profile* kernels). For ℓ and τ we use the published parameter settings. For σ we chose the best result among σ = {0.1,1,10,100,1000}.

performances than the string kernels on AA sequences as well as the non-substring kernels.

Conclusions

We have proposed new kernels that combine the benefits of physico-chemical descriptors for amino acids with the ones of string kernels. String kernels are powerful and expressive, yet one needs sufficiently many examples during training to learn relationships between amino acids in the very high dimensional space induced by the string kernel. Standard kernels based on physico-chemical descriptors of amino acids, on the other hand, cannot exploit the sequential structure of the input sequences and implicitly generate many more features, numerous of which will be biologically implausible. Here, one also needs more examples to learn which subset of features is needed for accurate discrimination, especially for longer protein sequences.

We could show that the proposed modifications of the WD kernel yield significant improvements in the prediction of MHC-binding peptides. As expected, the improvement is particularly strong when data is less abundant. For protein remote homology detection AA property-enhanced kernels can also lead to significant performance improvements. For the most sophisticated kernels using blast or PSI-blast profiles, however, information about the similarities of AAs can already be derived from the profiles and the improvement is marginal.

Overall, our experiments demonstrate that the proposed kernels indeed lead to a better performance than string kernels and non-substring kernels. These improvements are not major, but consistent. It has to be noted that a big difference between the previously proposed kernels and the proposed kernels cannot be expected: The proposed kernels essentially work on subsets of the features of previously proposed kernels and the improvements that we observe mainly come from the SVM's degraded performance when including uninformative features (which typically is not very pronounced).

In summary, the proposed modifications, in particular the combination with the RBF AA substring kernel, consistently yield improvements without seriously affecting the computing time (except for the Spectrum-RBF kernel). In all formulations, the original string kernel formulation can be recovered by appropriately choosing σ. Hence, when σ is included in model selection, the performance of the proposed kernels should be at least as good as the original string kernels. We therefore believe that the proposed kernels should be preferred over the original formulations for any protein sequence classification task.

List of abbreviations used

AA: amino acid; MHC: major histocompatibility complex; SVM: support vector machine; WD: weighted degree.

Acknowledgements

This work was partly supported by Deutsche Forschungsgemeinschaft (SFB 685, project B1).

This article has been published as part of *BMC Bioinformatics* Volume 11 Supplement 8, 2010: Proceedings of the Neural Information Processing Systems (NIPS) Workshop on Machine Learning in Computational Biology (MLCB). The full contents of the supplement are available online at http://www.biomedcentral.com/1471-2105/11?issue=S8.

Author details

[1]Center for Bioinformatics, Eberhard-Karls-Universität, Sand 14, 72076 Tübingen, Germany. [2]Friedrich Miescher Laboratory of the Max Planck Society, Spemannstr. 39, 72076 Tübingen, Germany.

Authors' contributions

NCT and GR conceived and designed the project. NCT prepared the data, implemented the kernels, performed experiments and drafted the manuscript. CW performed the MTL experiments and contributed to the preparation of manuscript. OK contributed to the discussion and helped writing the manuscript. GR supervised the project, implemented and performed experiments and contributed to the manuscript.

Competing interests

The authors declare that they have no competing interests.

Published: 28 April 2011

References

1. Saigo H, Vert JP, Ueda N, Akutsu T: **Protein homology detection using string alignment kernels.** *Bioinformatics* 2004, **20(11)**:1682-9.
2. Kuang R, Ie E, Wang K, Wang K, Siddiqi M, Freund Y, Leslie C: **Profile-based string kernels for remote homology detection and motif extraction.** *Proceedings IEEE Computational Systems Bioinformatics Conference* 2004.
3. Weston J, Leslie C, Ie E, Zhou D, Elisseeff A, Noble WS: **Semi-supervised protein classification using cluster kernels.** *Bioinformatics* 2005, **21(15)**:3241-3247.
4. Rätsch G, Sonnenburg S, Srinivasan J, Witte H, Müller KR, Sommer RJ, Schölkopf B: **Improving the Caenorhabditis elegans genome annotation using machine learning.** *PLoS Comput Biol* 2007, **3(2)**:e20.
5. Schweikert G, Zien A, Zeller G, Behr J, Dieterich C, Ong CS, Philips P, De Bona F, Hartmann L, Bohlen A, Krüger N, Sonnenburg S, Ratsch G: **mGene: accurate SVM-based gene finding with an application to nematode genomes.** *Genome Res* 2009, **19(11)**:2133-43.
6. Schultheiss SJ, Busch W, Lohmann JU, Kohlbacher O, Rätsch G: **KIRMES: kernel-based identification of regulatory modules in euchromatic sequences.** *Bioinformatics* 2009, **25(16)**:2126-33.
7. Roth V, Fischer B: **Improved functional prediction of proteins by learning kernel combinations in multilabel settings.** *BMC Bioinformatics* 2007, **8(Suppl 2)**:S12.
8. Ong CS, Zien A: **An Automated Combination of Kernels for Predicting Protein Subcellular Localization.** *Proceedings of the 8th Workshop on Algorithms in Bioinformatics (WABI)* Lecture Notes in Bioinformatics, Springer; 2008, 168-179.
9. Jacob L, Vert JP: **Efficient peptide-MHC-I binding prediction for alleles with few known binders.** *Bioinformatics* 2008, **24(3)**:358-66.
10. Röttig M, Rausch C, Kohlbacher O: **Combining structure and sequence information allows automated prediction of substrate specificities within enzyme families.** *PLoS Comput Biol* 2010, **6**:e1000636.
11. Leslie C, Eskin E, Noble WS: **The Spectrum Kernel: A String Kernel For SVM Protein Classification.** *In Proceedings of the Pacific Symposium on Biocomputing* 2002, 564-575.
12. Rätsch G, Sonnenburg S: **Accurate Splice Site Detection for Caenorhabditis elegans.** In *Kernel Methods in Computational Biology.* MIT Press;B Schölkopf KT, Vert JP 2004:277-298.
13. Meinicke P, Tech M, Morgenstern B, Merkl R: **Oligo kernels for datamining on biological sequences: a case study on prokaryotic translation initiation sites.** *BMC Bioinformatics* 2004, **5(169)**.

14. Shen B, Bai J, Vihinen M: **Physicochemical feature-based classification of amino acid mutations.** *Protein Eng Des Sel* 2008, **21**:37-44.
15. Pfeifer N, Kohlbacher O: **Multiple Instance Learning Allows MHC Class II Epitope Predictions Across Alleles.** In *Algorithms in Bioinformatics. Volume 5251*. Lecture Notes in Computer Science, Springer; 2008:210-221.
16. Rangwala H, Karypis G: **Profile-based direct kernels for remote homology detection and fold recognition.** *Bioinformatics* 2005, **21(23)**:4239-4247.
17. Venkatarajan M, Braun W: **New quantitative descriptors of amino acids based on multidimensional scaling of a large number of physical—chemical properties.** *Journal of Molecular Modeling* 2001, **7**:445-453.
18. Ong CS, Zien A: **An Automated Combination of Kernels for Predicting Protein Subcellular Localization.** *Proceedings of the 8th Workshop on Algorithms in Bioinformatics (WABI)* Lecture Notes in Bioinformatics, Springer; 2008, 186-179.
19. **.** In *Advances in Kernel Methods: Support Vector Learning*. Cambridge, MA, USA: MIT Press;Schölkopf B, Burges CJC, Smola AJ 1999:.
20. Tung CW, Ho SY: **POPI: predicting immunogenicity of MHC class I binding peptides by mining informative physicochemical properties.** *Bioinformatics* 2007, **23(8)**:942-949.
21. Leslie C, Eskin E, Cohen A, Weston J, Noble WS: **Mismatch string kernels for discriminative protein classification.** *Bioinformatics* 2004, **20(4)**:467-476.
22. Leslie C, Eskin E, Weston J, Noble W: **Mismatch String Kernels for Discriminative Protein Classification.** *Bioinformatics* 2004, **20(4)**:467-476.
23. Peters B, Bui HH, Frankild S, Nielsen M, Lundegaard C, Kostem E, Basch D, Lamberth K, Harndahl M, Fleri W, Wilson SS, Sidney J, Lund O, Buus S, Sette A: **A Community Resource Benchmarking Predictions of Peptide Binding to MHC-I Molecules.** *PLoS Comput Biol* 2006, **2(6)**:e65.
24. Kawashima S, Ogata H, Kanehisa M: **AAindex: Amino Acid Index Database.** *Nucleic Acids Research* 1999, **27**:368-369.
25. Henikoff S, Henikoff JG: **Amino acid substitution matrices from protein blocks.** *Proceedings of the National Academy of Sciences of the United States of America* 1992, **89(22)**:10915-10919.
26. Clark RM, Schweikert G, Toomajian C, Ossowski S, Zeller G, Shinn P, Warthmann N, Hu TT, Fu G, Hinds DA, Chen H, Frazer KA, Huson DH, Schölkopf B, Nordborg M, Rätsch G, Ecker JR, Weigel D: **Common Sequence Polymorphisms Shaping Genetic Diversity in Arabidopsis thaliana.** *Science* 2007, **317(5836)**:338-342.
27. **The Leslie Lab - Software - String Kernels..** [http://cbio.mskcc.org/leslielab/software/string-kernels].
28. Sonnenburg S, Rätsch G, Henschel S, Widmer C, Behr J, Zien A, de Bona F, Binder A, Gehl C, Franc V: **The SHOGUN Machine Learning Toolbox.** *Journal of Machine Learning Research* 2010, **11(z)**:1799-1802.
29. Sonnenburg S, Zien A, Philips P, Rätsch G: **POIMs: positional oligomer importance matrices—understanding support vector machine-based signal detectors.** *Bioinformatics* 2008, **24(13)**:i6-14.

doi:10.1186/1471-2105-11-S8-S7
Cite this article as: Toussaint *et al.*: Exploiting physico-chemical properties in string kernels. *BMC Bioinformatics* 2010 **11**(Suppl 8):S7.