Decoding Sequence Classification Models for Acquiring New Biological Insights

Ulrich Bodenhofer[1], Andreas Kothmeier[1], Ingrid G. Abfalter[1], Carsten C. Mahrenholz[2], and Sepp Hochreiter[1]
[1]Institute of Bioinformatics, Johannes Kepler University, Linz, Austria
[2]Institute of Medical Immunology, Charité Medical School, Berlin, Germany

Abstract

Classifying biological sequences is one of the most important tasks in computational biology. In the last decade, *support vector machines (SVMs)* in combination with *sequence kernels* have emerged as a de-facto standard. These methods are theoretically well-founded, reliable, and provide high-accuracy solutions at low computational cost. However, obtaining a highly accurate classifier is rarely the end of the story in many practical situations. Instead, one often aims to acquire biological knowledge about the principles underlying a given classification task. SVMs with traditional sequence kernels do not offer a straightforward way of accessing this knowledge.

In this contribution, we propose a new approach to analyzing biological sequences on the basis of support vector machines with sequence kernels. We first extract *explicit pattern weights* from a given SVM. When classifying a sequence, we then compute a *prediction profile* by distributing the weight of each pattern to the sequence positions that match the pattern. The final profile not only allows assessing the importance of a position, but also determining for which class it is indicative. Since it is unfeasible to analyze profiles of all sequences in a given data set, we advocate using *affinity propagation (AP)* clustering to narrow down the analysis to a small set of typical sequences.

The proposed approach is applicable to a wide range of biological sequences and a wide selection of sequence kernels. To illustrate our framework, we present the *prediction of oligomerization tendencies of coiled coil proteins* as a case study.

General Data Analysis Pipeline

Sequence Classification Using Support Vector Machines

Support vector machines (SVMs) are well-established standard methods for classifying biological sequences. Advantages of SVMs [2,8]:

- Maximizing the margin between two classes → proven to be a *near-optimal learning strategy*.
- Optimization problem is convex and quadratic → *global solution exists* and *can be found efficiently*.
- Only depend on very few hyperparameters → *easier model selection*.
- Can be *applied to any kind of data;* all needed is a meaningful positive semi-definite comparison measure (the so-called *kernel*) → great advantage for sequences (*cannot always be cast into vectorial data*)

SVMs in a Nutshell. Consider training data $\{(x_i, y_i) \mid i = 1,\ldots,l\}$, where x_i are sequences and $y_i \in \{-1,+1\}$ are binary labels. Discriminant function of SVM:

$$f(x) = b + \sum_{i=1}^{l} \alpha_i \cdot y_i \cdot k(x_i, x),$$

x: new data item to be classified; α_i: weights determined by SVM training (Lagrange multipliers); $k(.,.)$: kernel function.

Sequence Kernels. Wide range available [9], many of which can be expressed as [1]

$$k(x, y) = \sum_{p \in P} N(p, x) \cdot N(p, y),$$

P: set of sequence patterns; $N(p,x)$: number of occurrences/matches of pattern p in sequence x. This formulation includes the well-known spectrum kernel [6], the mismatch kernel [5], and the spatial sample kernel [4]. To correct for varying sequence lengths, it is often useful to normalize the kernel [9]:

$$k(x, y) = \frac{\sum_{p \in P} N(p, x) \cdot N(p, y)}{\sqrt{\sum_{p \in P} N(p,x)^2} \cdot \sqrt{\sum_{p \in P} N(p,y)^2}}$$

Extraction of Pattern Weights

SVMs are often *black-box predictors*. For sequence kernels represented as above, we can reformulate the discriminant function as (left: unnormalized kernel; right: normalized kernel) [1]:

$$f(x) = b + \sum_{i=1}^{l} \alpha_i \cdot y_i \cdot \sum_{p \in P} N(p, x) \cdot N(p, x_i) \qquad f(x) = b + \sum_{i=1}^{l} \alpha_i \cdot y_i \cdot \frac{\sum_{p \in P} N(p, x) \cdot N(p, x_i)}{\sqrt{\sum_{p \in P} N(p, x)^2} \sqrt{\sum_{p \in P} N(p, x_i)^2}}$$

$$= b + \sum_{p \in P} N(p, x) \cdot \underbrace{\sum_{i=1}^{l} \alpha_i \cdot y_i \cdot N(p, x_i)}_{w(p)} \qquad = b + \frac{1}{\sqrt{\sum_{p \in P} N(p, x)^2}} \sum_{p \in P} N(p, x) \underbrace{\sum_{i=1}^{l} \frac{\alpha_i \cdot y_i \cdot N(p, x_i)}{\sqrt{\sum_{p \in P} N(p, x_i)^2}}}_{w(p)}$$

$w(p)$: *individual contribution of each pattern* p.

- Negative $w(p)$: pattern p is indicative for the negative class
- Positive $w(p)$: pattern p is indicative for the positive class
- The higher the absolute value, the clearer the tendency
- $w(p)$ around zero: pattern does not occur or is irrelevant for classification task

Generalization to position-specific variants of sequence kernels is possible, too [1].

Prediction Profiles

Pattern weights provide the analyst with valuable knowledge which, however, may be incomplete, obscured or even misleading:

- patterns may in fact be part of larger or more complex patterns that were not included in P;
- occurrences of patterns are dependent, but the weights do not take any dependencies into account.

Another reformulation of discriminant function [7]:

$$f(x) = b + \sum_{j=1}^{L} s_j = \sum_{j=1}^{L} \left(s_j - (-\frac{b}{L})\right),$$

L: length of sequence; s_j: contribution of j-th letter in the sequence — computed as an appropriate portion of the weights of patterns matching the sequence in position j.

The contributions s_j can be plot as a *prediction profile* along the sequence:

- Negative s_j: letter at position j is indicative for the negative class
- Positive s_j: letter at position j is indicative for the positive class
- The higher the absolute value, the clearer the tendency
- s_j around zero: letter at position j is irrelevant for classification task

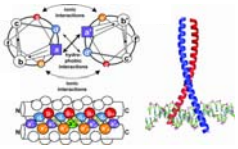
The values s_j can be plotted as a *profile along the sequence*. The discriminant function can be computed as the *area between the profile and the base line* $-b/L$.

Studying profiles of all sequences is unfeasible. We suggest to concentrate on a limited number of representative examples. To determine such *exemplars*, we recommend using *affinity propagation clustering* [3].

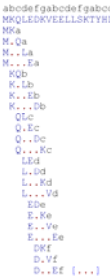
Case Study: Prediction of Oligomerization of Coiled Coils [7]

Introduction

Coiled coil: structural motif in which two or more α-helices are coiled together in a super-helical twist. Coiled coils are usually built of repeating patterns of amino acids, the so-called *heptad repeats*.



Our goal was to determine whether a given coiled coil segment tends to build a dimer (2 helices) or trimer (3 helices).



Classification Model

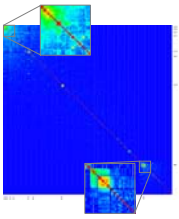
Data Preparation: The whole PDB was scanned with SOCKET to retrieve all coiled coil sequences with known 3D structure. We created a database of 385 dimeric and 92 trimeric coiled coil sequences with heptad registers (abcdefg) assigned by SOCKET. To augment this set with newly sequenced genome data, we employed a sophisticated BLAST approach with stringent filtering, which resulted in a combined dataset of 2043 dimers and 791 trimers.

Coiled Coil Kernel: We designed a novel kernel tailored to classification of coiled coil segments. It considers pairs of amino acids that are at most m positions apart and also takes the heptad positions of the residues into account (see left).

Model Selection: optimal model parameters were determined using nested cross-validation. Data were clustered such that training and test sequences had at most 60% sequence identity.

Pattern Weights

Pattern weights were computed from the final SVM as described above. A list of the 25 most dimeric and the 25 most trimeric sequence patterns is shown on the right hand side.

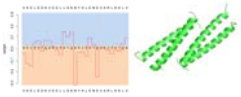


Clustering

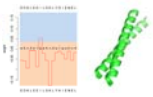
Negative (dimers) and positive class (trimers) were clustered by affinity propagation with respect to the coiled coil kernel to obtain a small number of representative exemplar sequences. The plot to the left shows a heatmap of sequence similarities arranged by the VAT algorithm with the eight most typical samples marked.

Prediction Profiles of 8 Typical Coiled Coils

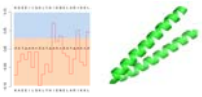
S355: GCN4 mutation Arg2Ser, Met3Val, Gln5Glu (1CE9)



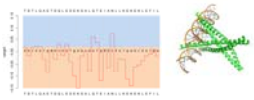
S315: GCN4 N terminus capped with acetamidobenzoic acid (1W5I)



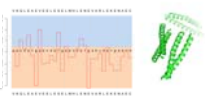
S333: GCN4 dimer asym. unit of four helix bundle mutation Leu9Ala (1UO5)



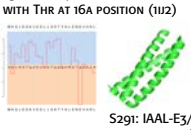
S371: Heterodimeric bZIP transcription factor c-Fos-c-Jun (1FOS)



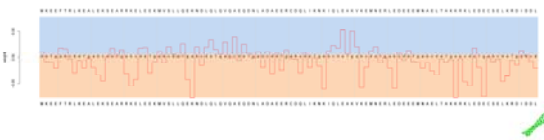
S364: GCN4 acid base heterodimer Leu12D, Leu16A (1KD9)



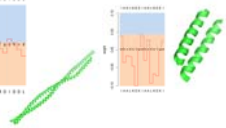
S36: GCN4-pVTL coiled-coil trimer with Thr at 16A position (1IJ2)



S391: Human beta-myosin S2 fragment (2FXM)



S291: IAAL-E3/K3 heterodimer (1UOI)



Availability

- ProCoil — R package and Web service for prediction and profiling of coiled coils: http://www.bioinf.jku.at/software/procoil/
- APCluster — R package for affinity propagation clustering: http://www.bioinf.jku.at/software/apcluster/

References

[1] U. Bodenhofer, K. Schwarzbauer, M. Ionescu, and S. Hochreiter. Modeling position specificity in sequence kernels by fuzzy equivalence relations. In J. P. Carvalho, D. Dubois, U. Kaymak, and J. M. C. Sousa, editors, *Proc. Joint 13th IFSA World Congress and 6th EUSFLAT Conference*, pages 1376–1381, Lisbon, July 2009.
[2] C. Cortes and V. N. Vapnik. Support vector networks. *Machine Learning*, **20**:273{297, 1986.
[3] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, **315**(5814):972{976, 2007.
[4] P. Kuksa, P.-H. Huang, and V. Pavlovic. A fast, large-scale learning method for protein sequence classification. In *8th Int. Workshop on Data Mining in Bioinformatics*, pages 29–37, Las Vegas, NV, 2008.
[5] C. Leslie, E. Eskin, A. Cohen, J. Weston, and W. S. Noble. Mismatch string kernels for discriminative protein classification. Bioinformatics, **1**(1):1–10, 2003.
[6] C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: a string kernel for SVM protein classification. In R. B. Altman, A. K. Dunker, L. Hunter, K. Lauderdale, and T. E. D. Klein, editors, *Pacific Symposium on Biocomputing*, pages 566–575. World Scientific, 2002.
[7] C. C. Mahrenholz, I. G. Abfalter, U. Bodenhofer, R. Volkmer, and S. Hochreiter. Complex networks govern coiled coil oligomerization: predicting and profiling by means of a machine learning approach. (submitted)
[8] B. Schölkopf and A. J. Smola. *Learning with Kernels. Adaptive Computation and Machine Learning*. MIT Press, Cambridge, MA, 2002.
[9] B. Schölkopf, K. Tsuda, and J.-P. Vert, editors. *Kernel Methods in Computational Biology*. MIT Press, Cambridge, MA, 2004.