SKINK: a web server for string kernel based kink prediction in α-helices

Tim Seifert1, Andreas Lund2, Benny Kneissl3,*, Sabine C. Mueller4, Christofer S. Tautermann5 and Andreas Hildebrandt1

1Department of Software Engineering and Bioinformatics, Johannes Gutenberg University of Mainz, 55128 Mainz, 2Institute for Computer Science, Johann Wolfgang Goethe-University Frankfurt am Main, 60054 Frankfurt am Main, 3Data Science, Pharma Research and Early Development Informatics (pREDi), Roche Diagnostics GmbH, 82377 Penzberg, 4Department of Human Genetics, Saarland University Faculty of Medicine, 66421 Homburg and 5Lead Identification and Optimization Support, Boehringer-Ingelheim Pharma GmbH & Co. KG, 88397 Biberach, Germany

ABSTRACT

Motivation: The reasons for distortions from optimal α-helical geometry are widely unknown, but their influences on structural changes of proteins are significant. Hence, their prediction is a crucial problem in structural bioinformatics. Here, we present a new web server, called SKINK, for string kernel based kink prediction. Extending our previous study, we also annotate the most probable kink position in a given α-helix sequence.

Availability and implementation: The SKINK web server is freely accessible at http://biows-inf.zdv.uni-mainz.de/skink. Moreover, SKINK is a module of the BALL software, also freely available at www.ballview.org.

Contact: benny.kneissl@roche.com

Received on November 11, 2013; revised on January 31, 2014; accepted on February 7, 2014

1 INTRODUCTION

In our previous study (Kneissl et al., 2011), we presented a new method for predicting kinks from amino acid sequences of α-helices implemented in the Biochemical Algorithm Library (BALL; Hildebrandt et al., 2010). Using string kernel based support vector machines (SVM), we were able to predict kinks in ~80% of all cases correctly, exceeding recently reported accuracies of alternative approaches (Hall et al., 2009; Langelaan et al., 2010; Meruelo et al., 2011; Rigoutsos et al., 2003; Yohannan et al., 2004). To make our approach publicly available by providing a user-friendly interface, we developed the web server SKINK. Moreover, we annotate the most probable kink position and calculate the corresponding probability value described shortly in the Section 2. In consequence, the region around the identified residue can be treated as flexible, e.g. in Molecular Dynamics (MD) simulations, while the remainder of the helix can be restrained.

2 METHODS

The prediction (classification) of an input sequence as kinked or non-kinked is described in our previous study (Kneissl et al., 2011) in detail. In addition, the most probable position of a putative kink is computed. Therefore, we divided the original sequence into overlapping k-mers of all odd sizes between 7 and 15 amino acids. Let the kink be in the center position of the k-mer; this corresponds to one to two neighbored helical turns, which are taken into account. These fragments are then predicted with the corresponding support vector machine as kinked or non-kinked. Using a trapezoidal-shaped function, the probability pk of the kink is located at position s in the currently considered subsequence is calculated by

\[ p_k = \sum_{i=3}^{3} \sum_{j=-1}^{7} \left[ \frac{7-i}{2} \cdot k_{s(i),s(i)+j} \right] \]

where k_{a,b} is 1 if subsequence [a, b] is predicted as kinked and otherwise 0. Hence, a position is more likely to be the kink if it appears near the center position of the k-mers that are labeled as kinked. The calculated probabilities are then normalized by the maximum possible value at each position such that each amino acid in the input sequence is assigned with a value between 0 and 1. Because we are not interested in kinks located at the end of a helix, the first and last two turns (seven positions) are ignored. Finally, the maximum values are chosen to determine the kink positions. If there are multiple maxima, two are merged if they are closer than four positions, else they are annotated as two kinks.

Applying this kink annotation approach on our previously used training dataset (downloadable from the Web site), we are able to identify 85.6% of 339 correctly predicted kinks (out of 366) within one helical turn.

3 SKINK

The SKINK web server is kept deliberately simple but well-structured to maintain clarity and is geared to the abilities of the users (see Fig. 1). First, there is a page Documentation, where some information about the dataset format as well as the precomputed support vector machine models is given. A detailed description and some analysis results of the datasets can be found in the page Data Sets. Moreover, all datasets can be downloaded here.

The two main functionalities are provided in the pages Predict Kinks and Create Model.

Users can query the server via the input text field or upload a file containing all sequences, which shall be classified as kinked or non-kinked. We allow two different input formats: The sequence can be in plain text (one sequence per line) or in the FASTA format. In addition, they can upload a file containing
all sequences in one of these two formats. Before continuing, the user has to choose at least one of the precomputed models or upload his/her own one in the LIBSVM (Chang and Lin, 2011) format. All chosen support vector machines are then applied on each input sequence. The classification whether the sequence is kinked or not is quite fast (about 15 s), whereas the annotation of the kink position can be up to three times longer because of the computation of all feature vectors of the different subsequences. The results are presented in a new page, where all information (input sequences, chosen models, logfile) of the request is collected. For each model a subdirectory is created where the corresponding result file is stored. In this file, each input sequence is labeled as non-kinked (0) or kinked (1), whereas in the latter case the kink position and the probability value are given, too.

To create a new model, the user has to choose a training dataset, which can either be one of the provided ones on the Web site or an own dataset. In both cases, the BALL::ParameterFile format is required. Thereafter, he/she has to decide how many different string kernels and how many different LIBSVM parameter setups he/she wants to apply, which can be combined for each training run on the following setup page. Hence, for one dataset all generated support vector machines are located in a separate subdirectory. Clicking ‘continue’ shows the setup page. Here, the user can re-check the uploaded dataset and choose different parameters for the string kernel functions. Each setup can be named in a unique user-specified name. Finally, the LIBSVM parameters, e.g. for the cost function or the number of cross validation folds, can be set. Clicking ‘continue’ shows the result page, where for each fold a separate SVM is created. Moreover, it shows a file with all sequences and the corresponding predicted labels as well as a file containing a short analysis, e.g. the confusion matrix and prediction accuracies.

Funding: Boehringer Ingelheim Pharma GmbH & Co. KG.

Conflict of Interest: none declared.

REFERENCES