J Comput Aided Mol Des (2014) 28:831–839
DOI 10.1007/s10822-014-9763-x

H-DROP: an SVM based helical domain linker predictor trained with features optimized by combining random forest and stepwise selection

Teppei Ebina · Ryosuke Suzuki · Ryotaro Tsuji · Yutaka Kuroda

Received: 23 January 2014 / Accepted: 9 June 2014 / Published online: 26 June 2014

Abstract Domain linker prediction is attracting much interest as it can help identifying novel domains suitable for high throughput proteomics analysis. Here, we report H-DROP, an SVM-based Helical Domain linker pRediction using OPtimal features. H-DROP is, to the best of our knowledge, the first predictor for specifically and effectively identifying helical linkers. This was made possible first because a large training dataset became available from IS-Dom, and second because we selected a small number of optimal features from a huge number of potential ones. The training helical linker dataset, which included 261 helical linkers, was constructed by detecting helical residues at the boundary regions of two independent structural domains listed in our previously reported IS-Dom dataset. 45 optimal feature candidates were selected from 3,000 features by random forest, which were further reduced to 26 optimal features by stepwise selection. The prediction sensitivity and precision of H-DROP were 35.2 and 38.8 %, respectively. These values were over 10.7 % higher than those of control methods including our previously developed DROP, which is a coil linker predictor, and PPRODO, which is trained with un-differentiated domain boundary sequences. Overall, these results indicated that helical linkers can be predicted from sequence information alone by using a strictly curated training data set for helical linkers and carefully selected set of optimal features. H-DROP is available at http://domserv.lab.tuat.ac.jp

Keywords Feature selection · Random forest · Stepwise selection · Support vector machine · Domain boundary prediction · Structural domain · Helical linker · Machine learning

Abbreviations

PSSM	Position specific score matrix
SVM	Support vector machine
RF	Random forest
MDGI	Mean decrease Gini index
MDA	Mean decrease accuracy
OFC	Optimal feature candidate set
ISD	Independent structural domain: domains that would fold independently according to IS-Dom criteria
Domain linker	Domain boundary region separating two ISDs
Coil linker	Domain linker forming mostly random coils
Helical linker	Domain linker forming mostly helices

Introduction

Large proteins often contain several structural domains that can fold independently from the rest of the protein [1, 2].

Teppei Ebina and Ryosuke Suzuki contributed equally to this work.

Electronic supplementary material The online version of this article (doi:10.1007/s10822-014-9763-x) contains supplementary material, which is available to authorized users.

T. Ebina · R. Suzuki · R. Tsuji · Y. Kuroda (✉)
Department of Biotechnology and Life Science, Tokyo University of Agriculture and Technology, 12-24-16 Nakamachi, Koganei-shi, Tokyo 184-8588, Japan
e-mail: ykuroda@cc.tuat.ac.jp

Present Address:
T. Ebina (✉)
Brain Science Institute, RIKEN, 2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan
e-mail: teppei-ebina@brain.riken.jp

Structural domains are usually easier to express, purify or characterize by high-throughput methods than their full length protein counterparts [3, 4]. Methods for detecting structural domains based solely on sequence information are thus gaining practical significance in diverse areas of proteomics research [5, 6]. Predicting domain regions based on sequence similarity to structural domain templates would be a classical and straight forward approach [7]. The disadvantage of a template-based prediction is that one indeed needs a template dataset, and "novel" domain without sequence similarity to known domains can thus not be identified (at least in a naïve way). In contrast, one can first detect domain boundaries using a machine learning method and use this information to assign the corresponding domain regions [8–11]. The sequence features of domain boundaries, such as a high frequency of proline in coil linkers [8, 11–14], appear to be widely conserved [11, 15], and machine-learning predictors can thus identify novel domains with boundaries conserving the training's linker sequence characteristics. The efficiency of this approach has been corroborated by the experimental detection and characterization of several novel structural domains [5, 16].

Although domain boundaries form random coils (coil boundary) [11, 14], a significant number of boundaries contain helices (helical boundary) [14]. Coil boundaries are predicted with relatively high efficiency whereas the helical boundaries are more difficult to predict [9–11]. This might be because helical boundaries are less common and supervised machine learning tends to detect the most common characteristics, but not the uncommon ones [17]. Nevertheless, the accurate prediction of helical boundaries would be highly valuable in proteomics research as they are found in many proteins, such as the well known Hsp70 molecular chaperons [18].

In this study, we first constructed a novel dataset of "helical linkers" which are 4 or more residue long sequences forming helical structures and separate two independent structural domains (ISDs) [19]. Afterwards we developed a helical linker predictor based on a support vector machine (SVM) trained using optimal features for characterizing helical linkers. The optimal features for distinguishing helical linkers from non-helical linkers were identified by combining a random forest (RF) and stepwise features selection procedures [11]. We started with an initial set of 3,000 features and reduced it to 45 using the RF feature selection. Then, we carried out a stepwise selection which resulted in 26 optimal features. The performances of our helical linker prediction were significantly superior to those of a random guess and other linker prediction methods including our previously developed coil linker predictor, DROP [11]. These observations stress the efficiency of our two step feature selection protocol for SVM based helical linker prediction.

Methods

Helical linker dataset

We first defined a linker as a region located between two continuous ISDs and forming little or no hydrogen bonds and hydrophobic clusters with its neighbor ISDs as assessed from the protein's atomic coordinates (see Supplementary Methods for details). ISDs were derived from the IS-Dom dataset with default parameters [19]. A linker was defined as a helical linker when more than 70 % of its residues were classified as helices by DSSP [20] (DSSP codes H, I and G). We chose representative sequences with sequence identities less than 30 % using a single linkage clustering. The final helical linker dataset (DS-Helical) included 255 proteins containing 261 helical linkers.

In addition we constructed a control dataset independent from DS-Helical by using the same IS-Dom filter, but based on the domain boundary definition of DomainParser [21]. This dataset included 343 proteins containing 347 helical linkers, and sequence identity with DS-Helical was less than 30 %.

Vector encoding

Each residue was encoded into a 3,000-dimensional real-vector according to our previously reported protocol (Fig. 1a) [11]. First, 576 different feature values of a residue were assigned to a vector: 544 amino acid indices [22], 20 PSSM elements [23], 3 probabilities of secondary structures [24], 2 α-helix/β-sheet core indices [25], 1 length of sequential hydrophobic core [26], 1 expected contact order index [27], 3 domain/intra-domain helix/helical-linker propensity indices [12] and 2 linker likelihood scores [28] (see Supplemental Methods). Then, the 576 features were averaged over different window sizes (±5, ±10, ±15 or ±20) for taking into account the local and semi-local sequence information. In addition, 5 sequence complexity indices calculated by Shannon's Entropy [29], 100 amino acid composition indices, and 15 similarity scores in amino acid composition to domain or helical linker regions were assigned to the vector. As a result, a 3,000 dimensional vector was generated for each residue [((544 + 20 + 3 + 2 + 1 + 1 + 3 + 2 = 576 features) × (4 averaging windows +1 un-averaged, single residue window) = 2,880) + 5 + 100 + 15 = 3,000] (Fig. 1a).

Random forest feature selection

We first reduced the number of features using the mean decrease of the Gini index (MDGI) calculated by a RF (Fig. 1b) [11, 30] algorithm implemented with the R-programming language (R-Random Forest package [31],

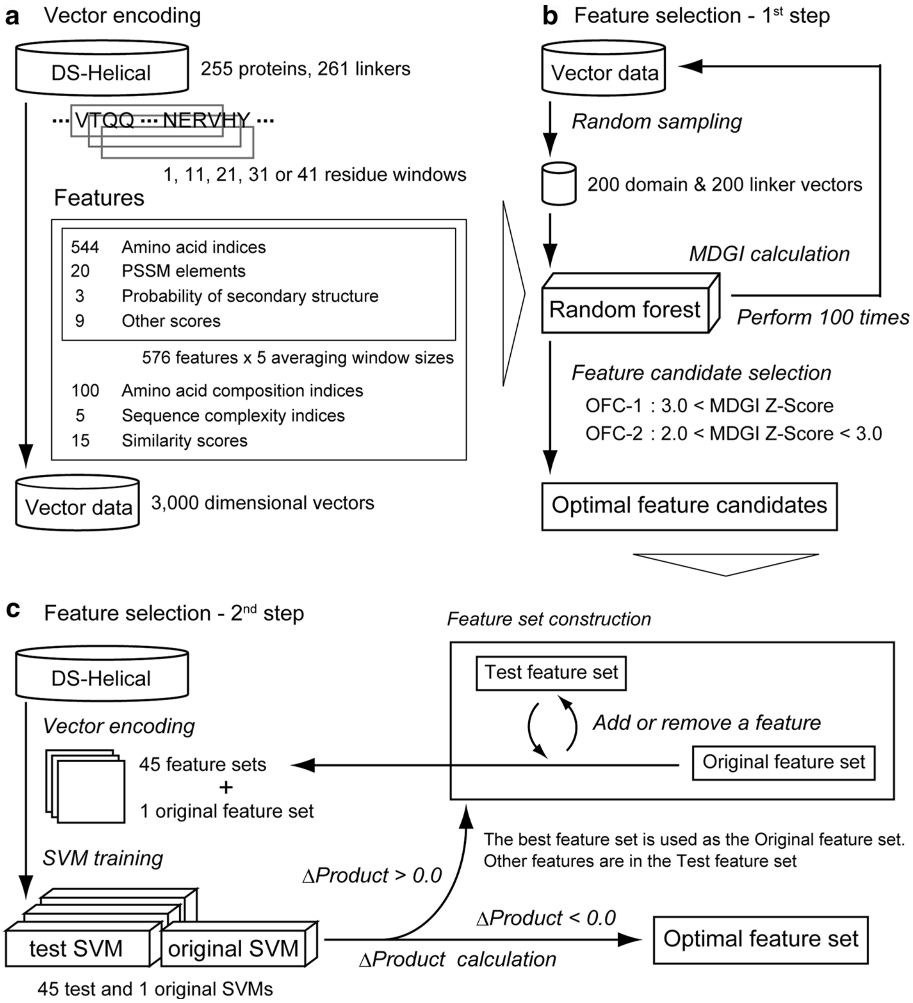
a Vector encoding
DS-Helical
255 proteins, 261 linkers
··· VTQQ ··· NERVHY ···
1, 11, 21, 31 or 41 residue windows
Features
544 Amino acid indices
20 PSSM elements
3 Probability of secondary structure
9 Other scores
576 features x 5 averaging window sizes
100 Amino acid composition indices
5 Sequence complexity indices
15 Similarity scores
Vector data
3,000 dimensional vectors
b Feature selection - 1st step
Vector data
Random sampling
200 domain & 200 linker vectors
MDGI calculation
Random forest
Perform 100 times
Feature candidate selection
OFC-1 : 3.0 < MDGI Z-Score
OFC-2 : 2.0 < MDGI Z-Score < 3.0
Optimal feature candidates
c Feature selection - 2nd step
DS-Helical
Vector encoding
45 feature sets
+
1 original feature set
SVM training
test SVM
original SVM
45 test and 1 original SVMs
Feature set construction
Test feature set
Add or remove a feature
Original feature set
The best feature set is used as the Original feature set.
Other features are in the Test feature set
ΔProduct > 0.0
ΔProduct < 0.0
ΔProduct calculation
Optimal feature set

Fig. 1 Scheme of the two-step feature selection. **a** Vector encoding: Each residue in the 255 multi-domain protein sequences was encoded into a vector using 3,000 features representing PSSMs, predicted secondary structures and other amino acid's physico-chemical properties. **b** Feature selection 1st Step: The 3,000 features were ranked according to their MDGI Z-score calculated using a random forest algorithm, and 45 features were defined as optimal feature candidates. The feature candidates were divided into OFC-1 and OFC-2, where 25 features were in OFC-1 and 20 features were in OFC-2. **c** Feature selection 2nd Step: One original and 45 test SVMs were constructed at each round of the stepwise selection. The original SVM was trained using the original feature set. The test SVM was trained using the original feature set, but a feature was removed from the set or added from the test feature set. A feature set yielding the highest ΔProduct was selected as the original feature set for the next round of selection. ΔProduct was calculated by subtracting the product of sensitivity and precision of the test SVM from that of original SVM. This process was repeated until ΔProduct of all test SVMs became negative. OFC-1 and OFC-2 were used as the original and test feature set in the first round, respectively

http://cran.r-project.org/). The MDGI values indicate the importance of the features for distinguishing residues in helical linkers from domain regions and might be useful for identifying the optimal feature candidates for helical linker prediction. DS-Helical contained many more domain residues than linker residues (66,117 and 4,136 residues, respectively). In order to reduce the computational time of the random forest feature selection and a potential over-fitting to a biased dataset, we reduced the number of domain and helical linker residues, by generating 100 sets of randomly selected 200 helical linker and 200 domain residues. We calculated the mean MDGI value by averaging 100 MDGI values for each of the 3,000 features. We chose potentially important features using the Z-Score of the mean MDGI (MDGI Z-Score) calculated as $\frac{xi-\bar{x}}{\sigma}$, where x_i is the mean MDGI of the feature i, and $\bar{x}$ and σ are its average and standard deviation of the mean MDGI over all of the 3,000 features. Features with MDGI Z-Score larger than 3.0 were classified as optimal feature candidate-1 (OFC-1) and with the score between 2.0 and 3.0 were classified as optimal feature candidate-2 (OFC-2).

Stepwise feature selection

The stepwise selection was started by using OFC-1 and OFC-2 as, respectively, the original and test sets of features (Fig. 1c). At each round of the stepwise selection we exhaustively compared the SVM predictors trained using the original feature set (original SVM) with a test SVM trained using the original set from which one feature was removed or the original set to which one feature was added (Fig. 1c). For each feature set, we optimized the training parameters using SVMLab (see subsection "Optimization of the SVM training parameters" for details), trained the SVM (described in subsection "Training of the SVM and prediction assessment"), and optimized the prediction parameters as described in subsection "Optimization of prediction parameters". The set that yielded the highest prediction performances was selected as the original set for the next round. This process was repeated until all test SVMs were less performing than the original SVM. The optimal features were defined as the features contained in the original set at the final round.

Optimization of the SVM training parameters

During the stepwise selection, we optimized the training parameters for each set of features using SVMLab (http://rubygems.org/gems/svmlab; SVMLab optimizes the parameters by maximizing Pearson's correlation coefficient between the input and output data). To reduce the computational time, we generated 5 sets of randomly selected 200 helical linker and non helical linker vectors, which SVMLab used for a five-fold cross validation of the parameters optimization. The optimal training parameters were determined as the respective median values.

Training of the SVM and prediction assessment

For each set of features, we trained an SVM^{light} [32] with an RBF kernel function with the above optimized parameters and using all of the helical and non-helical vectors. The training and prediction assessment were performed by using a five-fold cross validation test, so that test sequences had a maximum sequence similarity of 30 %. The helical linkers were predicted using the SVM output as follows: The raw SVM output values were smoothed using the optimal window size. The regions with smoothed values higher than the default threshold value were defined as helical linker candidates. The helical linker candidates located within 40 residues from the N- and C-terminal of a query sequence were excluded from our helical linker prediction. Finally, the linker candidate with the highest value in the sequence was predicted as a helical linker (first ranked prediction). The optimization of the smoothing window size and the default threshold value are described in subsection "Optimization of prediction parameters".

The prediction performances were assessed using standard definition for *sensitivity* and *precision* of the prediction: $Sensitivity = TP/(TP + FN)$; $Precision = TP/(TP + FP)$, where TP stands for True Positive (linker predicted as linker), FN means False Negative (linker predicted as non-linker), and FP is the False Positive (non linker predicted as linker). The prediction was defined as correct when the predicted helical linker residue with the highest SVM output value overlapped with a structure-defined linker residue [11].

Optimization of prediction parameters

The optimal smoothing window was determined by testing 9 window sizes (from ±0 to ±16 by step of 2 residues) and maximizing the product of the sensitivity and the precision with their default threshold value.

Residues with the smoothed SVM output values above and below a threshold value were classified as helical linker and domain residues, respectively. The default threshold value for each smoothing window size was determined so as to maximize $R_{TP} - R_{FP}$, where R_{FP} is the ratio of the number of domain residues predicted as helical linker residues to that of all domain residues, and R_{TP} is the ratio of the number of correctly predicted helical linker residues to that of all helical linker residues. R_{FP} and R_{TP} were calculated by varying threshold value from -14.0 to 14.0 by step of 0.01.

Result and discussion

Characteristics of helical linkers in DS-Helical

We first constructed a dataset of helical linkers containing 261 helical linkers (see Fig. 2). Residues at the linker termini had higher accessible surface area than domain residues, suggesting that the selected helical linkers act as a fairly solvent exposed and rigid linker for separating two ISDs. This tendency was reflected in the helical linker's amino acid composition. Although the amino acid composition of helical linkers differed only slightly from that of intra-domain α-helices (Fig. 3), helical linkers preferred hydrophilic residue (Lys and Arg) over hydrophobic ones (Leu, Ala, Ile, Val and Phe) [11, 14].

Optimal feature candidates selected by random forest

In the first step of the feature selection, we assessed the importance of the 3,000 features for the helical linker prediction using the MDGI Z-Score calculated during the random forest decision tree construction. From the initial

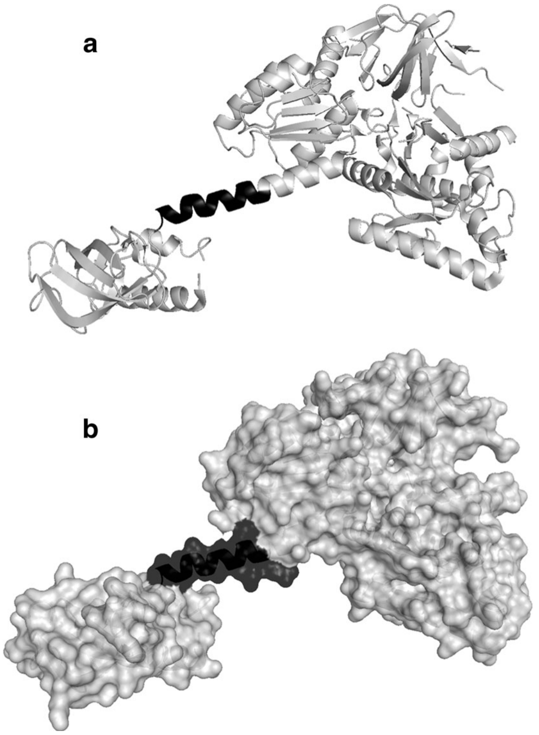

Fig. 2 Example of a helical linker in DS-Helical. **a** Ribbon model of a helical linker region in the translation initiation factor IF2/eIF5B (PDB ID: 1G7R chain A). The linker was located at residues 446–460. *White* and *black* regions indicate domain and helical linker regions, respectively. The average sequence lengths of helical linkers in DS-Helical were 15.8 ± 0.7. **b** Surface representation of the protein. Conventions are the same as in a. Images in (**a**) and (**b**) are created by Pymol (http://www.pymol.org/). The accessible surface area (ASA) of the helical linker residues was 55.1 ± 1.2 Å^2, which was slightly higher than the accessible surface area of residues in intra-domain helices, which was 45.4 ± 0.4 Å^2 (mean ± SEM). Similarly, residues at linker termini had higher ASAs than residues at domain termini (average ASA of 50.7 ± 2.5 Å^2 *vs* 30.3 ± 2.0 Å^2). The accessible surface area values were calculated by DSSP

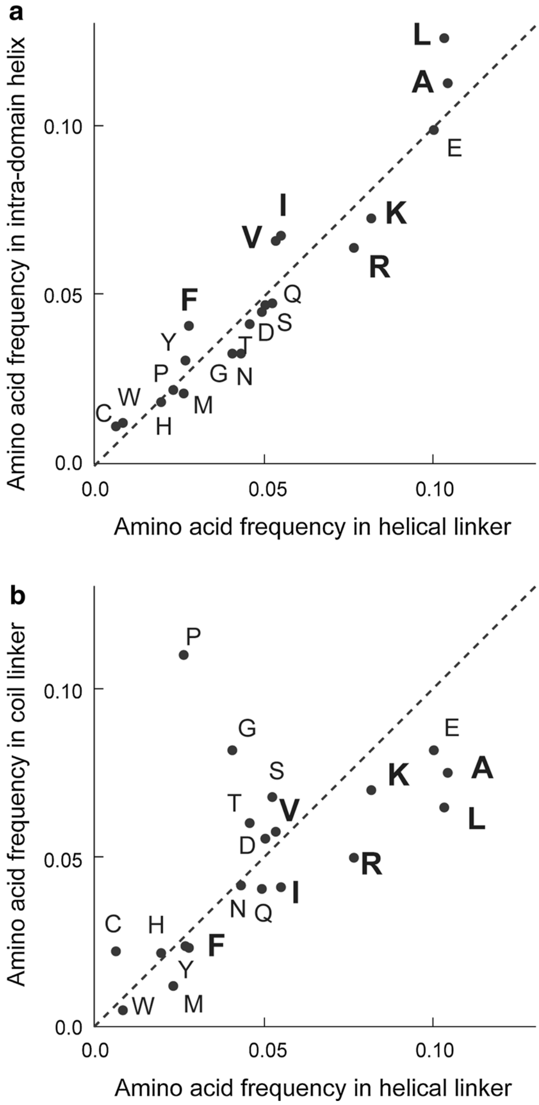

Fig. 3 Difference in amino acid frequency between helical linker and coil linker/intra-domain helix. **a** Amino acid compositions of helical linkers and intra-domain helices. The *vertical* and *horizontal axes* indicate the amino acid composition in intra-domain helices and helical linkers. The compositions were calculated using protein sequences in DS-Helical. Amino acids that are characteristic of helical linker and intra-domain helix are indicated by *bold letters*. **b** Amino acid compositions of helical and coil linkers. Similarly to helical linkers, coil linkers were obtained by selecting linkers in which 70 % or more of the residues were defined as non-helical and non-β-sheet by DSSP (DSSP codes B, S, T and blank), which yielded 764 coil linkers. The *vertical axis* indicates the composition in coil linkers, and other conventions were the same as in (**a**)

set of 3,000 features, 25 OFC-1 and 20 OFC-2 features were identified as the optimal feature candidates (Table S-1 and S-2, Fig. 4). The feature candidates were mostly related to the secondary structure formation. In particular, predicted secondary structure and amino acid composition similarity to helical linkers ranked higher than other features (Fig. 4). On the other hand, PSSM for Lys and Arg were not selected although both amino acids are preferred in helical linker regions (see above section). A similar loss of potentially important features was also observed in developing DROP, where only three predicted secondary structure were selected in OFC-1 [11].

In addition, we assessed the efficacy of the mean decrease accuracy (MDA), an importance score similar to the MDGI calculated by the random forest, for selecting optimal feature candidates. The Z-Scores calculated using MDA and MDGI were highly correlated, but more features had a MDA Z-Score > 2.0 (OFC-1′ and -2′) than when selected using corresponding the MDGI Z-Scores (OFC-1 and -2) (Fig. S-2). SVMs trained with feature candidates selected using the MDA had sensitivity and precision up to 4 % lower than SVMs trained using MDGI (Table 1).

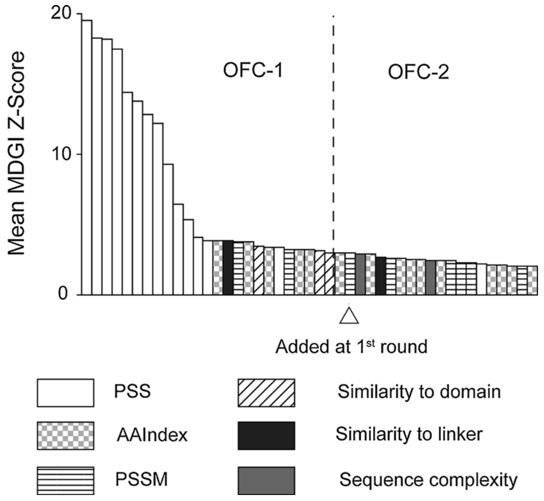

Fig. 4 MDGI Z-Score of features in OFC-1 and OFC-2. *Bars* indicate the mean MDGI Z-Score of features. *White arrow* represents the features added to the original feature set at the first round of the stepwise selection. The added feature was PSSM for Met. MDGI Z-Scores and the details of the features are listed in Supplementary Table S-1 and S-2

Thus, the MDGI appeared to be more efficient for selecting optimal feature candidates than the MDA.

Optimal feature combination selected by stepwise selection

Features that maximize the performances of the SVM helical linker prediction were selected from the optimal feature candidates in a stepwise way. The "optimal" feature set was reconstructed in a stepwise manner by adding a feature that increases the performances or removing a feature that decreases the performances at each round of the selection. Stepwise selection is not strictly an exhaustive search of the optimal feature combination, but it enables to estimate a sufficient number of combinations to yield one of the best set. We tested 46 feature combinations in each round (Fig. 1) and obtained 26 optimal features after two rounds of stepwise selection (Figs. 4, 5). Stepwise selection significantly improved the sensitivity and the precision of the SVM prediction from 29.9 to 35.2 % and from 32.9 to 38.8 %, respectively (Fig. 5; Table 1). Although the SVM was optimized using only the sensitivity and the precision of the prediction, and the *AUC* was thus not explicitly used during the stepwise selection, the *AUC* slightly improved from 0.806 to 0.816 (Fig. 6; assessed with DS-Helical). This corroborates our previous observations that small variations in *AUC* are sometimes reflected in substantial improvement of the final prediction performances. In turn, this observation motivated us to use the prediction's sensitivity and the precision for stepwise optimization [11].

Table 1 Dependence of the performances on initial feature set of stepwise selection

Predictor	Precision (%)	Sensitivity (%)
H-DROP		
Initial	32.9	29.9
Final	38.8	35.2
H-DROP-SD4[a]		
Initial	34.0	30.2
Final	37.2	34.8
H-DROP-SD8[b]		
Initial	27.6	24.1
Final	38.6	35.2
SD2-MDGI[c]	30.8	28.7
SD3-MDA[d]	31.1	29.1
SD2-MDA[e]	26.5	24.9

Initial and final indicates the performances of the respective original SVMs in the first and the final round of the stepwise selection

[a] H-DROP-SD4 was trained using 15 optimal features selected by the stepwise selection from OFC-1a and OFC-2a, in which the feature sets included 12 and 33 features with the MDGI Z-Score higher than 4.0 and between 2.0 and 4.0, respectively

[b] H-DROP-SD8 was trained using 11 optimal features selected from 9 OFC-1b and 36 OFC-2b features. OFC-1b and OFC-2b contained features with the MDGI Z-Score higher than 8.0 and between 2.0 and 8.0, respectively

[c] SD2-MDGI was trained using all of the features in OFC-1 and OFC-2

[d, e] SD3-MDA and SD2-MDA were trained using OFC-1′ and OFC-2′ features, respectively (see Sect. 3.2). OFC-1′ and OFC-2′ were constructed using the same procedure as that used for OFC-1 and -2, but the features were assessed based on their MDAs

The final SVM helical linker predictor was obtained as the original SVM at the second round of the stepwise selection. The predictor was trained using the 26 optimal features (Table S-1 and Fig. 4) with the SVM parameters *gamma, C* and *E* set to 0.67, 31.62 and 0.1, respectively. We also set the parameter *j* to 16.0, which is the cost factor for adjusting the training weight between the helical linker and domain residue vectors in DS-Helical. The raw SVM output values were smoothed using a 25-residue window and the default *threshold value* was set to -0.67.

In order to estimate the dependence of the prediction performances on the initial feature set of the stepwise selection, we constructed two additional helical linker predictors, H-DROP-SD4 and H-DROP-SD8 trained using 15 and 11 optimal features, respectively (Tables 1 and S-1). Among the three predictors, H-DROP demonstrated the highest performances (Table 1).

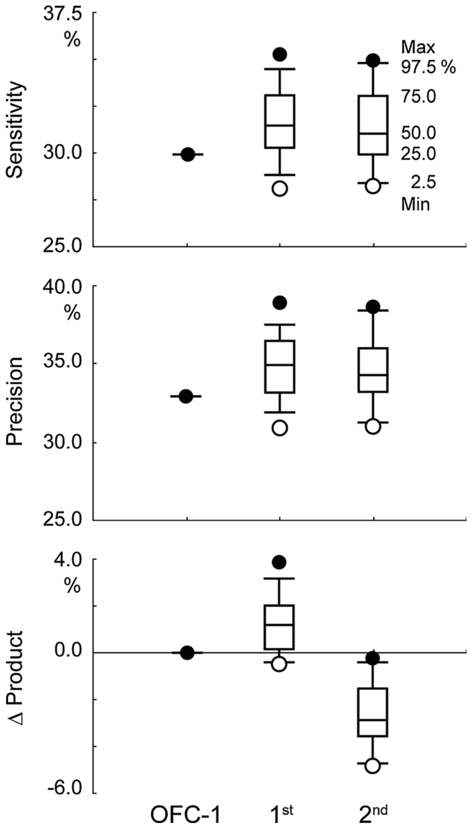

Fig. 5 Performance improvements during the stepwise selection. *Boxes* indicate the 2.5, 25, 50, 75 and 97.5th percentiles of the sensitivity (*top*), precision (*middle*) and ΔProduct (*bottom*) of the test SVMs at each round of the selection. *Black* and *white circles* indicate, respectively, the maximum and minimum values of the test SVMs at the rounds. The negative value of the maximum ΔProduct at the third round indicates that the prediction performances of the original SVMs are higher than those of all test SVMs. Thus the stepwise selection was terminated at round 2

Comparison to control predictors

We compared the prediction performances of our helical linker predictor, H-DROP, to those of control methods including DROP [11], our previously reported SVM based coil linker predictor. H-DROP's helical linker prediction sensitivity and precision were the highest (Table 2), even when the 40 termini residues were included in the calculation decreasing H-DROP performances by a mere 3 %. The performances of DROP were the lowest, and those of PPRODO [9], though higher than a random guess, were significantly lower than those of H-DROP (Table 2). Additionally, a naïve model that would predict helical linker using only the relative accessible area [33] and secondary structure prediction by PSIPRED [24] performed poorly (Tables 2, 3), indicating that carefully trained predictors are necessary to distinguish the faint characteristics of helical linkers.

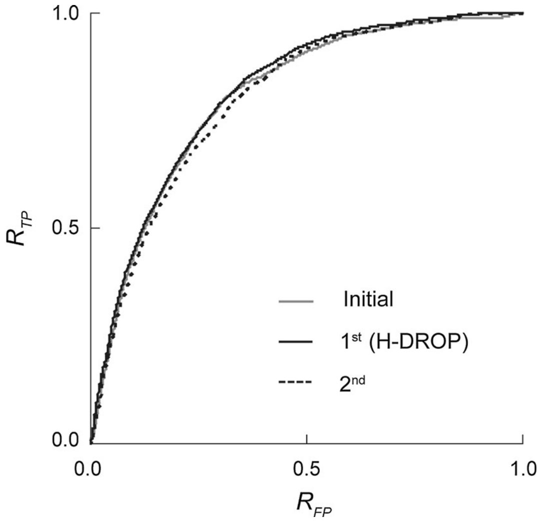

Fig. 6 ROC curves of the SVM predictors during stepwise selection. The ROC curves were calculated by five-fold cross validation tests of the SVM predictors with the highest performances at each round. The *AUC* values were 0.806, 0.816 and 0.803 for the SVMs at initial, 1st and 2nd round, respectively

In contrast to its high performance for predicting helical linkers, H-DROP did not perform well in predicting coil linkers. DROP and PPRODO were the highest and the second highest, respectively in predicting coil linkers (Table 2). Overall these results strongly suggest that DROP and H-DROP achieve the best prediction performance in their respective categories because they were trained using only coil and helical linkers, respectively. The sequence characteristics of helical and coil linkers are different (Fig. 3), and are difficult to detect simultaneously by simple two-class machine learning based predictors (Fig. 7). In addition, the inclusion of different types of linker regions in the training decreases the entire prediction performances. PPRODO, which is trained to detect domain boundaries without distinguishing its secondary structure, yielded significantly lower prediction performances than those of DROP and H-DROP in, respectively, coil and helical linker predictions (Table 2).

Ability to identify "novel" domain

To assess the ability of H-DROP to detect "novel" domains, we constructed a helical linker dataset independent from the training dataset by using DomainParser [21] for the initial domain boundary definition. The results confirmed the cross validation calculations reported in Table 2, and the prediction performances of H-DROP were significantly higher than those of other predictors (Table 3). This result suggests that sequences in DS-Helical are representative for the "real world protein", and that predictors trained with it should be able to predict "novel" domains.

Table 2 Prediction performances for helical and coil linkers

Predictor	Helical linker		Coil linker	
	Precision (%)	Sensitivity (%)	Precision (%)	Sensitivity (%)
H-DROP	38.8	35.2	5.2	3.4
H-DROP[a]	35.5	33.0	–	–
H-DROP-NoPDB	33.7	31.4	–	–
H-DROP-NoBLAST	28.1	25.3	–	–
PPRODO	25.2	24.5	21.7	20.4
DROP	10.9	10.7	29.1	27.7
ACCPro + PSIPRED	14.3	14.1	–	–
Random	12.6	12.2	5.7	6.1
Random[a]	7.96	7.96	–	–

The predictions were performed with our linker datasets including 247 and 764 helical and coil linkers (see legend of Fig. 3 for coil linker definition). All values were calculated using the first ranked prediction. [a] Control values calculated using all residues in the sequence, i.e., by including the 40 residue at both termini (The statistics are little influenced as only 9 linkers were predicted in the 40-residue protein's termini). H-DROP-NoPDB and H-DROP-NoBLAST are test calculations computed, respectively, by excluding PDB sequences from the PSSM calculation and by fully removing PSSM from the prediction (see Results and Discussion for details). The performances of H-DROP for the helical linkers were calculated using a five-fold cross validation test. The predictions of PPRODO were performed using its downloadable package with the default parameters. The performances of PPRODO were calculated using the results of the second network prediction [13]. The performances of DROP were calculated using the first-ranked predictions. The coil linkers used for DROP training were excluded from this test prediction (when all coil linkers were used, the sensitivity and the precision were 29.1 and 30.7 %, respectively). The ACCPro + PSIPRED is a naïve predictor prepared using a combination of predicted relative solvent accessibility by ACCPro [33] and the results of secondary structure predication by PSIPRED. A predicted helix with the highest accessibility was predicted as helical linker, and the residue at the center of the helical linker was used for computing the prediction performances. When two or more predicted helices had the same accessibility, the linker closer to the center of the query sequence was predicted as a helical linker. Random indicates values calculated by randomly selecting one residue from the protein sequence and checking whether it belongs to the helical linker. The selection was performed 100 times for each protein and the results were averaged

Table 3 Prediction performances with an independent dataset

Predictor	Precision (%)	Sensitivity (%)
H-DROP	29.4	28.2
DROP	5.8	5.7
PPRODO	17.2	17.0
ACCPro + PSIPRED	8.7	8.6
Random	11.4	11.3

The dataset contained 347 novel helical linkers that were not included in DS-Helical. The dataset was constructed in the same way as DS-Helical but ISDs were derived using DomainParser boundary definition. The maximum sequence identity between DS-Helical and this dataset was 27.1 %

We also examined how H-DROP would perform for sequences without sequence similarity to known sequences. To this, we derived two control predictors from H-DROP, and assessed their prediction performances. First, we constructed H-DROP-NoBLAST by setting PSSM vector elements to 0.0 in H-DROP, and using PSIPRED values calculated using BLOSUM62 matrix instead of the PSSM. Second, we constructed H-DROP-NoPDB, which used PSSM computed by removing all PDB sequences from the search (Genebank nr) sequences. The two predictors performed worse than H-DROP but still better than the other control predictors (Table 2), again suggesting the ability of H-DROP to detect domains in novel sequences. Finally, the prediction performances calculated with the "novel" sequence dataset (Table 3) further corroborated the efficacy of H-DROP for predicting helical linkers in novel protein targets.

Though domain prediction from sequence remains imperfect, it is nevertheless becoming a standard tool for experimentalists that dissect a novel protein into structurally and often functionally stable units. This has been corroborated by experiments in which our previously developed PASS domain predictor [34], and domain linker predictors (DROP [11] and DLP-SVM [10]) were used to successfully dissect large proteins into their domains and enabled their experimental characterization (see refs. 3,5,16 for example).

70% of domain linkers form coils, but a large minority (over 20 % according to our estimate based on PDB structures) contain helices longer than 4 residues, which have sequential characteristics different from coil linkers (Fig. 3). H-DROP might thus be a "niche" predictor, but it is the only one that specifically aims at detecting helical linkers, and we believe that it will contribute to the arsenal of bioinformatics tools for experimentalists willing to dissect large proteins into domains that are readily characterized using high throughput methods.

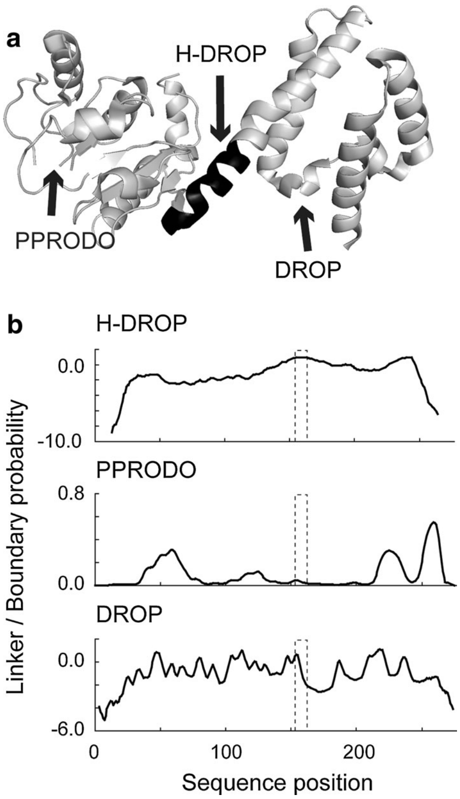

Fig. 7 Example of a helical linker prediction. **a** Ribbon model of hypothetical protein TM1727 (PDB ID: 2I76 chain A) which contains a helical linker at residues 151–164 (*black region* in the model). *Arrows* indicate the predicted domain boundary or linker by PPRODO (residue 60), DROP (residue 218) and H-DROP (residue 161). **b** Domain boundary and linker probability value. *Solid lines* indicate probability values calculated by H-DROP (*top*), PPRODO (*middle*) and DROP (*bottom*). *Broken lines* represent the structually defined helical linker region. The *horizontal axes* indicate the residue number in the sequence. The output values of H-DROP showed clear differences from those of DROP and PPRODO. Output values of H-DROP had peaks at the structure-defined helical linker regions whereas those of DROP and PPRODO did not

Conclusion

In conclusion, we developed a novel helical linker prediction method using an SVM trained with optimal features selected by random forest and stepwise feature selection. Although the sequence characteristics of helical linkers might be weaker than those of coil linkers, our two step feature selection identified optimal features in a realistic computational time (~70h on our 8 Xeon Linux server). H-DROP predicted helical linkers, which are not predicted by other methods, suggesting that H-DROP will be a useful for finding novel structural domains.

Acknowledgments We thank Mr. Yuki Umezawa for providing the IS-Dom data set, Ms. Ewa Mikiciuk for English proof reading, and members of the Kuroda's lab at TUAT for discussion and technical assistance. This work is supported by a Grant-in-aid from the Japanese Society for the Promotion of Science to Y.K. (JSPS, KAKENHI-18500225) and a JST-NBDC grant to T.E.

References

1. Brenner SE (2000) Nat Struct Biol 7(Suppl):967–969
2. Ekman D, Bjorklund AK, Frey-Skott J, Elofsson A (2005) J Mol Biol 348(1):231–243
3. Hondoh T, Kato A, Yokoyama S, Kuroda Y (2006) Protein Sci 15(4):871–883
4. Chikayama E, Kurotani A, Tanaka T, Yabuki T, Miyazaki S, Yokoyama S, Kuroda Y (2010) BMC Bioinformatics 11:113
5. Vastermark A, Almen MS, Simmen MW, Fredriksson R, Schioth HB (2011) BMC Evol Biol 11:123
6. Jacobs SA, Podell ER, Cech TR (2006) Nat Struct Mol Biol 13(3):218–225
7. Xue Z, Xu D, Wang Y, Zhang Y (2013) Bioinformatics 29(13):i247–256
8. Miyazaki S, Kuroda Y, Yokoyama S (2002) J Struct Funct Genomics 2(1):37–51
9. Sim J, Kim SY, Lee J (2005) Proteins 59(3):627–632
10. Ebina T, Toh H, Kuroda Y (2009) Biopolymers 92(1):1–8
11. Ebina T, Toh H, Kuroda Y (2011) Bioinformatics 27(4):487–494
12. Suyama M, Ohara O (2003) Bioinformatics 19(5):673–674
13. Dumontier M, Yao R, Feldman HJ, Hogue CW (2005) J Mol Biol 350(5):1061–1073
14. George RA, Heringa J (2002) Protein Eng 15(11):871–879
15. Miyazaki S, Kuroda Y, Yokoyama S (2006) BMC Bioinformatics 7:323
16. Hasegawa J, Tokuda E, Tenno T, Tsujita K, Sawai H, Hiroaki H, Takenawa T, Itoh T (2011) J Cell Biol 193(5):901–916
17. Forman G (2003) J Mach Learn Res 3:1289–1305
18. Zhu X, Zhao X, Burkholder WF, Gragerov A, Ogata CM, Gottesman ME, Hendrickson WA (1996) Science 272(5268): 1606–1614
19. Ebina T, Umezawa Y, Kuroda Y (2013) J Comput Aided Mol Des 27(5):419–426
20. Kabsch W, Sander C (1983) Biopolymers 22(12):2577–2637
21. Guo JT, Xu D, Kim D, Xu Y (2003) Nucleic Acids Res 31(3):944–952
22. Kawashima S, Pokarowski P, Pokarowska M, Kolinski A, Katayama T, Kanehisa M (2008) Nucleic Acids Res 36(Database issue):D202–205
23. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Nucleic Acids Res 25(17):3389–3402
24. Jones DT (1999) J Mol Biol 292(2):195–202
25. Chou PY, Fasman GD (1978) Adv Enzymol Relat Areas Mol Biol 47:45–148
26. Coeytaux K, Poupon A (2005) Bioinformatics 21(9):1891–1900
27. Garbuzynskiy SO, Lobanov MY, Galzitskaya OV (2004) Protein Sci 13(11):2871–2877
28. Tanaka T, Yokoyama S, Kuroda Y (2006) Biopolymers 84(2):161–168
29. Shenkin PS, Erman B, Mastrandrea LD (1991) Proteins 11(4):297–313
30. Wang M, Zhao XM, Takemoto K, Xu H, Li Y, Akutsu T, Song J (2012) PLoS ONE 7(8):e43847
31. Liaw A, Wiener M (2002) R News 2(3):18–22
32. Joachims T (1999) Making large-scale SVM learning practical. In: Schölkopf B, Burges C, Smola A (eds) Advances in Kernel methods: support vector learning. MIT, Cambridge
33. Cheng J, Randall AZ, Sweredoski MJ, Baldi P (2005) Nucleic Acids Res 33(Web Server issue):W72–76
34. Kuroda Y, Tani K, Matsuo Y, Yokoyama S (2000) Protein Sci 9(12):2313–2321