Probabilistic variable-length segmentation of protein sequences for discriminative motif mining (DiMotif) and sequence embedding (ProtVecX)

Ehsaneddin Asgari[1,2], **Alice McHardy**[2], **and Mohammad R.K. Mofrad**[1,3,*]

[1]Department of Bioengineering, University of California, Berkeley, CA 94720, USA
[2]Computational Biology of Infection Research, Helmholtz Centre for Infection Research, Brunswick 38124, Germany
[3]Molecular Biophysics and Integrated Bioimaging, Lawrence Berkeley National Lab, Berkeley, CA 94720, USA
[*]mofrad@berkeley.edu

ABSTRACT

In this paper, we present peptide-pair encoding (PPE), a general-purpose probabilistic segmentation of protein sequences into commonly occurring variable-length sub-sequences. The idea of PPE segmentation is inspired by the byte-pair encoding (BPE) text compression algorithm, which has recently gained popularity in subword neural machine translation. We modify this algorithm by adding a sampling framework allowing for multiple ways of segmenting a sequence. PPE can be inferred over a large set of protein sequences (Swiss-Prot) and then applied to a set of unseen sequences. This representation can be widely used as the input to any downstream machine learning tasks in protein bioinformatics. In particular, here, we introduce this representation through protein motif mining and protein sequence embedding. (i) DiMotif: we present DiMotif as an alignment-free discriminative motif miner and evaluate the method for finding protein motifs in different settings. The significant motifs extracted could reliably detect the integrins, integrin-binding, and biofilm formation-related proteins on a reserved set of sequences with high F1 scores. In addition, DiMotif could detect experimentally verified motifs related to nuclear localization signals. (ii) ProtVecX: we extend k-mer based protein vector (ProtVec) embedding to variable-length protein embedding using PPE sub-sequences. We show that the new method of embedding can marginally outperform ProtVec in enzyme prediction as well as toxin prediction tasks.
Availability: Implementations of our method will be available under the Apache 2 licence at http://llp.berkeley.edu/dimotif and http://llp.berkeley.edu/protvecx.

1 Introduction

Bioinformatics and natural language processing (NLP) are research areas that have greatly benefited from each other since their beginnings and there have been always methodological exchanges between them. Levenshtein distance [1] and Smith–Waterman [2] algorithms for calculating string or sequence distances, the use of formal languages for expressing biological sequences [3, 4], training language model-based embeddings for biological sequences [5], and using state-of-the-art neural named entity recognition architecture [6] for secondary structure prediction [7] are some instances of such influences. Similar to the complex syntax and semantic structures of natural languages, certain biophysical and biochemical grammars dictate the formation of biological sequences. This assumption has motivated a line of research in bioinformatics to develop and adopt language processing methods to gain a deeper understanding of how functions and information are encoded within biological sequences [4, 5, 8]. However, one of the apparent differences between biological sequences and many natural languages is that biological sequences (DNA, RNA, and proteins) often do not contain clear segmentation boundaries, unlike the existence of tokenizable

words in many natural languages. This uncertainty in the segmentation of sequences has made overlapping k-mers one of the most popular representations in machine learning for all areas of bioinformatics research, including proteomics [5, 9], genomics [10, 11], and metagenomics [12, 13]. However, it is unrealistic to assume that fixed-length k-mers are units of biological sequences and that more meaningful units need to be introduced. Although in some sequence-labeling tasks (e.g. secondary structure prediction or binding site prediction) sequences are implicitly divided into variable-length segments as the final output, methods to segment sequences into variable-length meaningful units as inputs of downstream machine learning tasks are needed. We recently proposed nucleotide pair encoding for phenotype and biomarker detection in 16S rRNA data [14], which is extended in this work for protein informatics.

Here, we propose a segmentation approach for dividing protein sequences into frequent variable-length sub-sequences, called peptide-pair encoding (PPE). We took the idea of PPE from byte pair encoding (BPE) algorithm, which is a text compression algorithm introduced in 1994 [15]that has been also used for compressed pattern matching in genomics [16]. Recently, BPE became a popular word segmentation method in machine translation in NLP for vocabulary size reduction, which also allows for open-vocabulary neural machine translation [17]. In contrast to the use of BPE in NLP for vocabulary size reduction, we used this idea to increase the size of symbols from 20 amino acids to a large set of variable-length frequent sub-sequences, which are potentially meaningful in bioinformatics tasks. In addition, as a modification to the original algorithm, we propose a probabilistic segmentation in a sampling framework allowing for multiple ways of segmenting a sequence into sub-sequences. In particular, we explore the use of PPE for protein sequence motif mining as well as training embeddings for protein sequences.

Motif mining: In biological sequences (DNA, RNA, and proteins), motifs are short sub-sequences that are presumed to have important biological functions; examples of such patterns are transcription factor binding sites, splice junctions, recruiting enzyme sites, and protein–protein interaction sites [18]. Motif mining has been one of the prominent problems in bioinformatics research. Protein motifs can be either short linear motifs (SLiMs) or larger in some case (e.g. Zinc finger) [19, 20]. Various methods have been proposed for finding protein motifs. These methods mostly framed as finding motifs in a set of similar sequences and usually benefit from sequence alignment algorithms [21], including (but not limited to) MEME Suite [22], QSLiMFinder [23],SLiMSearch [24], and DoReMi [25] as well as hidden Markov model (HMM)-based approaches (e.g., HH-MOTiF [26] and Phylo-HMM [27]). Although most motif mining methods are limited to finding continuous patterns because of their computational complexity, some methods find gapped motifs as well, which is closer to the real scenario [18]. Most of the traditional approaches look for motifs in a set of positive sequences. However, since other randomly conserved patterns may also exist in such sequences, reducing the false positive rate is a challenge for motif mining [28]. In order to address this issue, some studies have incorporated the use of negative samples to increase both the sensitivity and specificity of motif mining. Instances of such approaches are DEME [29] (using a Bayesian framework over alignment columns), discriminative HMM [30, 31],and DLocalMotif [32] (using position and entropy information), as well as deep-learning approaches [33]. General-purpose or specialized datasets are dedicated to maintaining a set of experimentally verified motifs from various resources (e.g., gene ontology). ELM [34] as a general-purpose dataset of SLiM, and NLSdb [35] as a specialized database for nuclear-specific motifs are instances of such efforts. Evaluation of mined motifs can be also subjective. Since the extracted motifs do not always exactly match the experimental motifs, residue-level or site-level evaluations have been proposed [26]. Despite great effort in this area, computational motif mining has remained a challenging task and the state-of-the-art *de novo* approaches have reported relatively low precision and recall scores, even at the residue level [26].

Protein embedding: Word embedding has been one of the revolutionary concepts in NLP over the recent years and has been shown to be one of the most effective representations in NLP [36, 37, 38]. In

particular, skip-gram neural networks combined with negative sampling [39] has resulted in state-of-the-art performance in a broad range of NLP tasks [38]. Recently, we introduced k-mer-based embedding of biological sequences using skip-gram neural network and negative sampling [5], which became popular for protein feature extraction and has been extended for various classifications of biological sequences [40, 41, 42, 43, 44, 45].

In this work, inspired by unsupervised word segmentation in NLP, we propose a general-purpose segmentation of protein sequences in frequent variable-length sub-sequences called PPE, as a new representation for downstream machine learning tasks. This segmentation is trained once over large protein sequences (Swiss-Prot) and then is applied on a given set of sequences. In this paper, we use this representation for developing a protein motif mining framework as well as protein sequence embedding.

(i) **DiMotif**: We suggest a discriminative and alignment-free approach for motif mining that is capable of finding multi-part motifs. We do not use sequence alignment; instead, we propose the use of general-purpose segmentation of positive and negative input sequences into PPE sequence segments. Subsequently, we used statistical tests to identify the significant discriminative features associated with the positive class, which are our ultimate output motifs. At the end, we create sets of multi-part motifs using information theoretic measures on the occurrence patterns of motifs on the positive set. We evaluate a shortlist of extracted motifs on the classification of reserved sequences of the same phenotype for integrins, integrin-binding proteins, and biofilm formation proteins, where the phenotype has been detected to have a high F1 score. We also evaluate the performance of our method for finding experimentally verified nuclear localization signal (NLS) motifs. However, a detailed analysis of the motifs and their biophysical properties are beyond the scope of this study, as the main focus is on introducing the method.

(ii) **ProtVecX**: We extend our previously proposed protein vector embedding (Protvec) [5] trained on k-mer segments of the protein sequences to a method of training them on variable-length segments of protein sequences, called ProtVecX. We evaluate our embedding via three protein classification tasks: (i) toxin prediction (binary classification), (ii) subcellular location prediction (four-way classification), and (iii) prediction of enzyme proteins versus non-enzymes (binary classification). We show that concatenation of the raw k-mer distributions with the embedding representations can improve the sequence classification performance over the use of either of k-mers only or embeddings only. In addition, combining of ProtVecX with k-mer occurrence can marginally outperform the use of our originally proposed ProtVec embedding together with k-mer occurrences in toxin and enzyme prediction tasks.

2 Material and Methods

2.1 Datasets

Motif mining datasets

Integrin-binding proteins: We extracted two positive and negative lists for integrin-binding proteins using the gene ontology (GO) annotation in the UniProt database [46]. For the positive class, we selected all proteins annotated with the GO term GO:0005178 (integrin-binding). Removing all redundant sequences resulted in 2966 protein sequences. We then used 10% of sequences as a reserved set for evaluation and 90% for motif mining and training purposes. For the negative class, we selected a list of proteins sequences which were annotated with the GO term GO:0005515 (protein binding), but they are annotated as neither integrin-binding proteins (GO:0005178) nor the integrin complex (GO:0008305). Since the resulting set was still large, we limited the selection to reviewed Swiss-Prot sequences and filtered redundant sequences, resulting in 20,117 protein sequences, where 25% of these sequences (5029 sequences) were considered as the negative instances for training and validation, and 297 randomly selected instances (equal to 10% of the positive reserved set) as the negative instances for the negative part of the reserved set.

Integrin proteins: We extracted a list of integrin proteins from the UniProt database by selecting entries annotated with the GO term GO:8305 (integrin complex) that also had integrin as part of their entry name. Removing the redundant sequences resulted in 112 positive sequences. For the negative sequences, we selected sequences which were annotated for transmembrane signaling receptor activity (to be similar to integrins) (GO:4888) but which were neither the integrin complex (GO:8305) nor integrin-binding proteins (GO:0005178). Selection of reviewed Swiss-Prot sequences and removal of redundant proteins resulted in 1155 negative samples. We used 10% of both the positive and negative sequences as the reserved set for evaluation and 90% for motif mining and training purposes.

Biofilm formation: Similar to integrin-binding proteins, positive and negative lists for biofilm formation were extracted via theri GO annotation in UniProt [46]. For the positive class, we selected all proteins annotated with the GO term GO:0042710] (biofilm formation). Removing all redundant sequences resulted in 1450 protein sequences (90% identity). For the negative class, we selected a list of protein sequences annotated within the parent node of biofilm formation in the GO database that were classified as being for multi-organism cellular process (GO:44764) but not biofilm formation. Since the number of resulting sequences waslarge, we limited the selection to reviewed Swiss-Prot sequences and filtered the redundant sequences, resulting in 1626 protein sequences. Again, we used 10% of both the positive and negative sequences as a reserved set for evaluation and 90% for motif mining and training purposes.

Nuclear localization signals: We used the NLSdb dataset containing nuclear export signals and NLS along with experimentally annotated nuclear and non-nuclear proteins [35]. By using NLSdb annotations from nuclear proteins, we extracted a list of proteins experimentally verified to have NLS, ending up with a list of 416 protein sequences. For the negative class, we used the protein sequences in NLSdb annotated as being non-nuclear proteins. NLSdb also contains a list of 3254 experimentally verified motifs, which we used for evaluation purposes.

Protein classification datasets

Sub-cellular location of eukaryotic proteins: We used the TargetP 4-classes dataset of sub-cellular locations: (i) 371 mitochondrial proteins, (ii) 715 pathway or signal peptides, (iii) 1214 nuclear proteins, and (iv) 438 cytosolic protein sequences [47], where the redundant proteins were removed.

Toxin prediction: We used the toxin dataset provided by ToxClassifier [48]. The positive set contained 8093 protein sequences annotated in Tox-Prot as being animal toxins and venoms [49]. For the negative class, we chose the 'Hard' setting of ToxClassifier [48], where the negative instances were 7043 protein sequences in UniProt which were not annotated in Tox-Prot but were similar to Tox-Prot sequences to some extent.

Enzyme detection: We downloaded two lists of enzyme and non-enzyme proteins (22,168 protein sequences per class) provided by the 'NEW' dataset of Deepre [50].

2.2 Peptide-pair encoding

PPE training

The input to the PPE algorithm is a set of sequences and the output would be segmented sequences and segmentation operations, an ordered list of amino acid merging operations to be applied for segmenting new sequences. At the beginning of the algorithm, we treated each sequence as a list of amino acids. As detailed in Algorithm 1, we then searched for the most frequently occurring pair of adjacent amino acids in all input sequences. In the next step, the selected pairs of amino acids were replaced by the merged version of the selected pair as a new symbol (a short peptide). This process was continued until we could not find a frequent pattern or we reached a certain vocabulary size (Algorithm 1).

In order to train a general-purpose segmentation of protein sequences, we trained the segmentation

Data: $Seqs$ = Set of Swiss-Prot protein sequences, f = minimum number of sequences containing the newly emerged symbol
Result: S = Divided sequences into variable sub-sequences, $Merge_opt$ = merging operations
$Sym = \{A,H,K,T,E,C,V,N,W,Y,F,Q,G,P,D,L,S,R,M,I\}$;
S = list of $Seqs$, where each sequence is a list of symbols $\in Sym$;
$Merge_opt = stack()$;
$SymbFreq$ = mapping symbol pairs in S to their frequencies;
$f_{current}$ = max frequency in $SymbFreq$;
while $f < f_{current}$ **do**
  sym1, sym2 = argmax ($SymbFreq$);
  S = merge all consecutive sym1 & sym2 into $< sym1, sym2 >$ in S;
  $Sym.push(< sym1, sym2 >)$;
  $Merge_opt.push(sym1, sym2)$;
  update($SymbFreq$);
  $current_f$ = max frequency in $SymbFreq$;
end

Algorithm 1: Adapted Byte-pair algorithm (BPE) for segmentation of protein sequences

over the most recent version of the Swiss-Prot database [51], which contained 557,012 protein sequences. We continued the merging steps for T iterations of Algorithm 1, which ensured that we captured the motifs present with a minimum frequency f in all Swiss-Prot sequences (we set the threshold to a minimum of $f = 10$ times, resulting in $T \approx 1$ million iterations). Subsequently, the merging operations can be applied to any given protein sequences as a general-purpose splitter.

Monte Carlo PPE segmentation

The PPE algorithm for a given vocabulary size (which is analogous to the number of merging steps in the training) divides a protein sequence into a unique sequence of sub-sequences. Further merging steps result in enlargement of sub-sequences, which results in having fewer sub-sequences. Such variations can be viewed as multiple valid schemes of sequence segmentation. For certain tasks, it might be useful to consider a protein sequence as a chain of residues and, in some cases, as a chain of large protein domains. Thus, sticking to a single segmentation scheme will result in ignoring important information for the task of interest. In order to address this issue, we propose a sampling framework for estimating the segmentation of a sequence in a probabilistic manner. We sample from the space of possible segmentations for both motif mining and embedding creation.

Different segmentation schemes for a sequence can be obtained by a varying number of merging steps (N) in the PPE algorithm. However, since the algorithm is trained over a large number of sequences, a single merging step will not necessarily affect all sequences, and as we go further with merging steps, fewer sequences are affected by the newly introduced symbol. We estimated the probability density function of possible segmentation schemes with respect to N by averaging the segmentation alternatives over 1000 random sequences in Swiss-Prot for $N \in [10000, 1000000]$, with a step size of 10000. For each N, we counted the average number of introduced symbols relative to the previous step; the average is shown in Figure 1. We used this distribution to draw samples from the vocabulary sizes that affected more sequences (i.e. those introducing more alternative segmentation schemes). To estimate this empirical distribution with a theoretical distribution, we fitted a variety of distributions (Gaussian; Laplacian; and

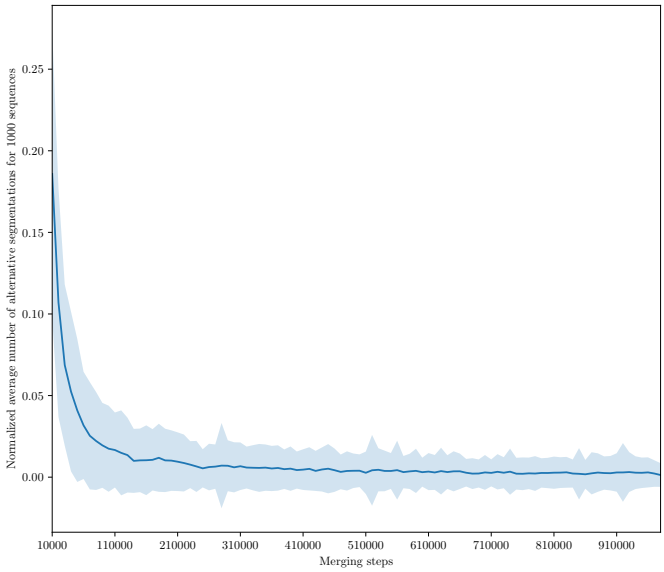

Figure 1. Average number of segmentation alternation per merging steps for 1000 Swiss-prot sequences.

Alpha, *Beta*, and *Gamma* distributions) using maximum likelihood and found the *Alpha* that fitted the empirical distribution the best. Subsequently, we use the fitted Alpha distribution to draw segmentation samples from a sequence in a Monte Carlo scheme. Consider $\Phi_{i,j}$ as the $l1$ normalized "bag-of-word" representations in sequence i, using the jth sample from the fitted *Alpha* distribution. Thus, if we have M samples, the "bag-of-sub-sequences" representation of the sequence i can be estimated as $\Phi_i = \frac{1}{M}\sum_j^M \Phi_{i,j}$.

For detecting sequence motifs, we represented each sequence as an average over the count distribution of M samples of segmentation (M=100) drawn from the *Alpha* distribution. The alternative is to use only the vocabulary size (e.g., the median of *Alpha*), referred to as the single segmentation scheme in this paper.

2.3 DiMotif protein sequence motif mining

Our proposed method for motif detection, called DiMotif, finds motifs in a discriminative setting via PPE features. We used the false discovery rate-corrected χ^2 test ($\alpha = 0.05$) to identify the most significant discriminative features between the positive and negative classes. Since we were looking for sequence motifs related to the positive class, we filtered the selected significant features based on their direction of correlation to ensure we excluded motifs related to the negative class.

We segmented both the training and test datasets with the trained PPE segmentation in Swiss-Prot (§2.2). We then find the significant discriminative motifs for integrins, integrin-binding proteins, biofilm formation, and NLS proteins by using a χ^2 test of the training dataset.

Evaluation of integrins, integrin-binding proteins, and biofilm formation motifs: In order to evaluate the obtained motifs, we trained linear support vector machine classifiers over the training data, but only used motifs related to the positive class among the top 1000 motifs as well as a short list of features. Next, we tested the predictive model on a reserved test set. Since the training and testing sets are disjoint, the classification results were indications of motif quality. We used both probabilistic and non-probabilistic segmentation methods to obtain PPE representations of the sequences. We report the precision, recall, and F1 of each classifier's performance.

Evaluation of NLS motifs: In the case of NLS motifs, we used the list of 3254 experimentally or manually verified motifs from NLSdb. Thus, in order to evaluate our extracted motifs, we directly

compared our motifs with those found in earlier verifications. Since for long motifs, finding exact matches is challenging, we report three metrics, the number of motifs with at least three consecutive amino acid overlaps, the number of sequences in the baseline that had a hit with more than 70% overlap (A to B and B to A), and finally the number of exact matches. In addition to Swiss-Prot-based segmentation, in order to see the effect of a specialized segmentation, we also trained PPE segmentation over a set of 8421 nuclear protein sequences provided by NLSdb [35] and performed the same evaluation.

2.3.1 Kulback–Leibler divergence to find multi-part motifs

As discussed in§ 1, protein motifs can be multi-part patterns, which is ignored by many motif-finding methods. In order to connect the separate parts, we propose to calculate the symmetric Kullback–Leibler (KL) divergence [52] between motifs based on their co-occurences in the positive sequences as follows:

$$D_{\mathrm{KL}_{\mathrm{sym}}}(M_p \| M_q) = \sum_{i}^{N} M_p(i) \log \frac{M_p(i)}{M_q(i)} + M_q(i) \log \frac{M_q(i)}{M_p(i)},$$

where M_p and M_q are, respectively, the normalized occurrence distributions of motif p and motif q across all positive samples and N is the number of positive sequences. Next, we used the condition of $(D_{\mathrm{KL}_{\mathrm{sym}}} = 0)$ to find co-occurring motifs splitting the motifs into equivalence classes. Each equivalent class indicates a multi-part or a single-part motif. Since we considered a ”bag of motifs”, the parts of multi-part motifs are allowed to be far from each other in the primary sequence.

Motif visualization: For visualization purposes DiMotif clusters motifs based on their co-occurrences in the positive class by using hierarchical clustering over the pairwise symmetric KL divergence. The motifs are then colored based on the most frequent secondary structure they assume in the sequences in the Protein Data Bank (PDB). For each motif, it visualizes their mean molecular weight, mean flexibility [53], mean instability [54], mean surface accessibility [55], mean kd hydrophobicity [56], and mean hydrophilicity [57] with standardized scores (zero mean and unit variance), where the dark blue is the lowest and the dark red is the highest possible value (e.g. Figure 3).

Secondary structure assignment

Using the trained segmentation over the Swiss-Prot sequences, we segmented all 385,937 protein sequences in the current version of the PDB [58], where their secondary structure was provided. By segmenting all secondary structures at the same positions as the corresponding sequences, we obtained a mapping from each sequence segment to all its possible secondary structures in the PDB. We used this information in the visualization of motifs (see Figure 3).

2.4 ProtVecX: Extended variable-length protein vector embeddings

We trained the embedding on segmented sequences obtained from Monte Carlo sampling segmentation on the most recent version of the Swiss-Prot database [51], which contains 557,012 protein sequences. Since this embedding is the extended version of ProtVec, we call it ProtVecX. As explained in §2.2 we segmented each sequence with the vocabulary size samples drawn from an $Alpha$ distribution. This ensured that we considered multiple ways of segmenting sequences during the embedding training. Subsequently, we trained a skip-gram neural network for embedding on the segmented sequences [39]. The skip-gram neural network is analogous to language modeling, which predicts the surroundings (context) for a given textual unit (shown in Figure 2). The skip-gram's objective is to maximize the following log-likelihood:

$$\sum_{t=1}^{M} \sum_{c \in [t-N, t+N]} \log p(w_c \mid w_t), \tag{1}$$

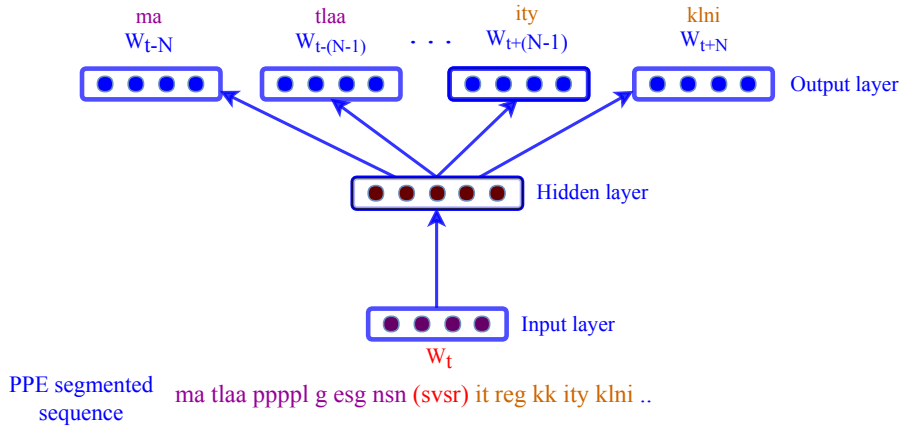

Figure 2. Skip-gram neural network for training language model-based embedding. In this framework the inputs are the segmented sequences and the network is trained to predict the surroundings PPE units.

where N is the surrounding window size around word w_t, c is the context indices around index t, and M is the corpus size in terms of the number of available words and context pairs. We parameterize this probability of observing a context word w_c given w_t by using word embedding:

$$p(w_c \mid w_t; \theta) = \frac{e^{v_c \cdot v_t}}{\sum_{c' \in \mathscr{C}} e^{v_{c'} \cdot v_t}}, \tag{2}$$

where $\mathscr{C}$ denotes all existing contexts in the training data. However, iterating over all existing contexts is computationally expensive. This issue can be efficiently addressed by using negative sampling. In a negative sampling framework, we can rewrite Equation 1 as follows:

$$\sum_{t=1}^{T} \left[\sum_{c \in [t-N, t+N]} \log\left(1 + e^{-s(w_t,\, w_c)}\right) + \sum_{w_r \in \mathscr{N}_{t,c}} \log\left(1 + e^{s(w_t,\, w_r)}\right) \right], \tag{3}$$

where $\mathscr{N}_{t,c}$ denotes a set of randomly selected negative examples sampled from the vocabulary collection as non-contexts of w_t and $s(w_t,\, w_c) = {v_t}^{\top} \cdot v_c$ (parameterization with the word vector v_t and the context vector v_c). For training embeddings on PPE units, we used the sub-word level skip-gram, known as fasttext [59]. Fasttext embedding improves the word representations by taking character k-mers of the sub-words into consideration in calculating the embedding of a given word. For instance, if we take the PPE unit *fggagvg* and $k = 3$ as an example, it will be represented by the following character 3-mers and the whole word, where '<' and '>' denote the start and the end of a PPE unit:

$\mathscr{S}_{fggagvg}$={`<fg', `fgg', `gga', `gag', `agv', `gvg', `vg>', `<fggagvg>'}

In the fasttext model, the scoring function will be based on the vector representation of k-mers ($2 \leq k \leq 6$) that exist in textual units (PPE units in this case), $s(w_t, w_c) = \sum_{x \in \mathscr{S}_{w_t}} v_x^{\top} v_c$.

We selected a vector dimension of 500 for the embedding (v_t's) and a window size of 20 (similar to ProtVec [5]). A k-mer-based ProtVec of the same vector size and the same window size trained on Swiss-Prot was used for comparison.

2.4.1 Embedding-based classification

For the classification, we used a Multi-Layer-Perceptrons neural network architecture with five hidden layers using Rectified Linear Unit as the nonlinear activation function. We used the softmax activation function at the last layer to produce the probability vector that could be regarded as representing posterior probabilities. To avoid overfitting, we performed early stopping and also used dropout at hidden layers. As baseline representations, we used k-mers, ProtVec [5], PPE vectors, and their combinations. For both ProtVec and ProtVecX, the embedding of a sequence is calculated as the summation of its k-mers or PPE unit vectors. We evaluated these representation in three protein classification tasks: (i) toxin prediction (binary classification) with the 'Hard' setting in the ToxClassifier database [48], (ii) subcellular location prediction (four-way classification) using the dataset provided by TargetP [47], and (iii) prediction of enzyme proteins versus non-enzymes (binary classification) using the NEW dataset [50].

3 Results

3.1 Sequence motifs and evaluation results

Integrins, integrin-binding proteins, and biofilm formation proteins: The trained classifiers over the extracted motifs associated with the positive class could reliably predict the reserved integrins, integrin-binding proteins, and biofilm formation proteins with F1 scores of 0.89, 0.89, and 0.75 respectively (Table 3). For integrin and biofilm formation, the probabilistic segmentation helps in predictions of the reserved dataset. Picking a smaller fraction of positive class motifs still resulted in a high F1 for the test set. For biofilm formation, the probabilistic segmentation improved the classification F1 score from 0.72 to 0.73 when only 48 motifs were used, where single segmentation even using more features obtained an F1 score of 0.70 (Table 3). The classification results suggest that the motifs mined from the training set are general enough to detect proteins of a similar type in the test set.

A detailed serious biophysical investigation of the extracted motifs is beyond the scope of this study. However, as an example, for integrin-binding proteins, the RGD motif, the most well-known integrin-binding motif was among the most significant motifs in our approach [60, 61, 62]. Other known integrin-binding motifs were also among the most significant ones, such as RLD [61], KGD (the binding site for the $\alpha II\beta 3$ integrins [63]), GPR (the binding site for $\alpha_x\beta 2$ [62]), LDT (the binding site for $\alpha_4\beta 7$ [62]), QIDS (the binding site for $\alpha_4\beta 1$ [62]), DLLEL (the binding site for $\alpha_v\beta 6$ [62]), [tldv,rldvv,gldvs] (similar motifs to LDV, the binding site for $\alpha_4\beta 1$ [60]), rgds [64], as well as the PEG motif [65].

Examples of DiMotif visualization output for the top extracted motifs are provided for interested readers, related to integrin-binding proteins (Figure 3), biofilm formation (Figure 4), and integrin complexes (Figure 5).

Nuclear localization signal: as shown in Table 2, the probabilistic segmentation has been more successful in detecting experimentally verified motifs. Training the segmentation on nuclear proteins resulted in slightly better but still competitive results with using general-purpose Swiss-Prot segmentation. A full list of multi-part extracted NLS motifs is provided as the supplementary material for follow up studies.

3.2 Results of protein classification tasks using embedding

Concatenation of embeddings with raw kmer distributions improves the classification performance (Table 3). This result suggest that k-mers can be more specific than embeddings in protein classifications. However, embbedings can provide complementary information to the k-mers and improve the classification performance. Concatenation of 3-mers with either of ProtVecX or ProtVec embedding performs very competitively and even for sub-cellular prediction task ProtVec performs slightly better. However, concatenation of 3-mers with the ProtVecX result in higher F1 in enzyme classification and toxin protein

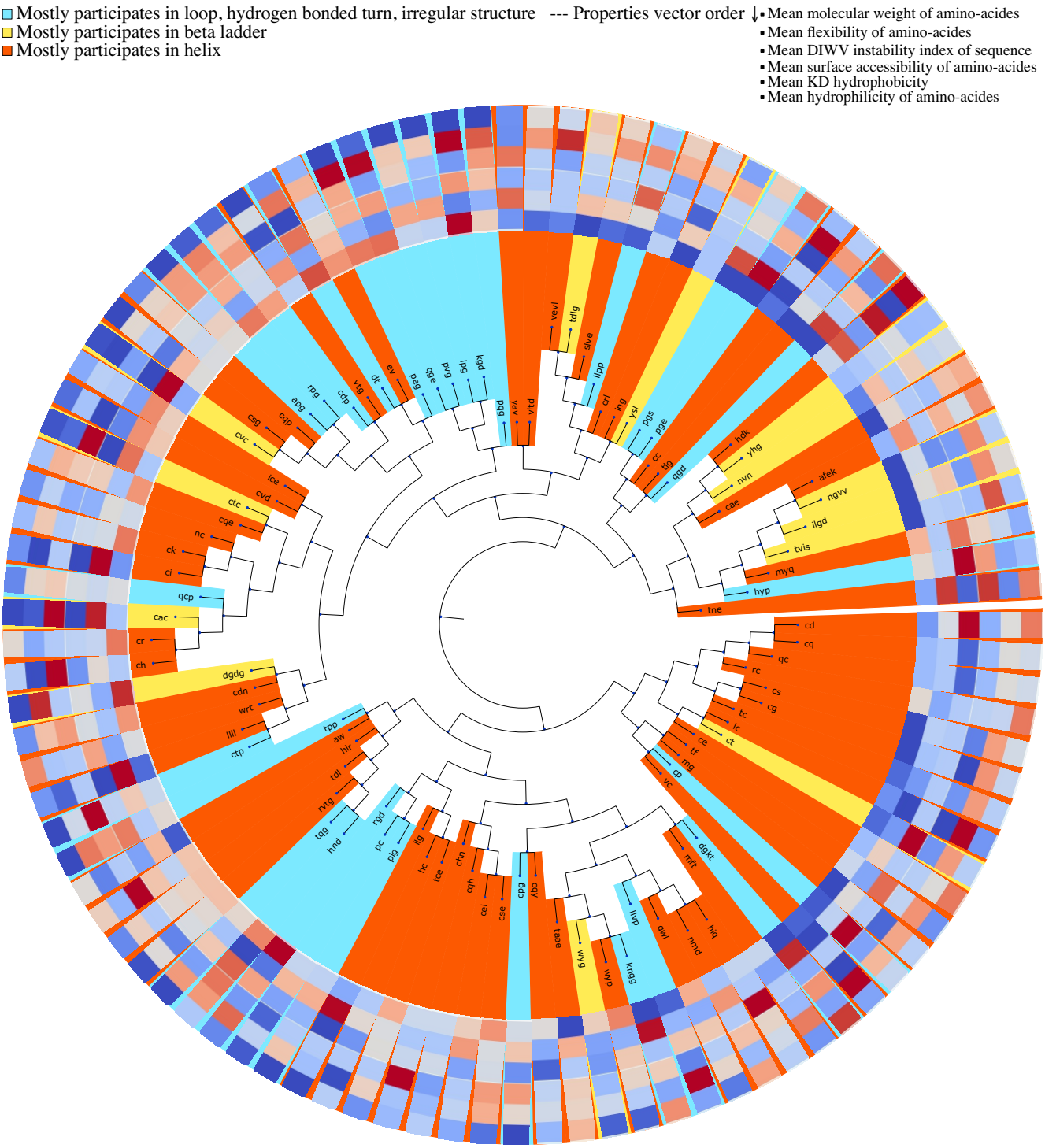

Figure 3. Clustering of integrin-binding-specific motifs based on their occurrence in the annotation proteins. Each motif is colored based on the most frequent secondary structure it assumes in the Protein Data Bank (PDB) structure out of all PDB sequences.

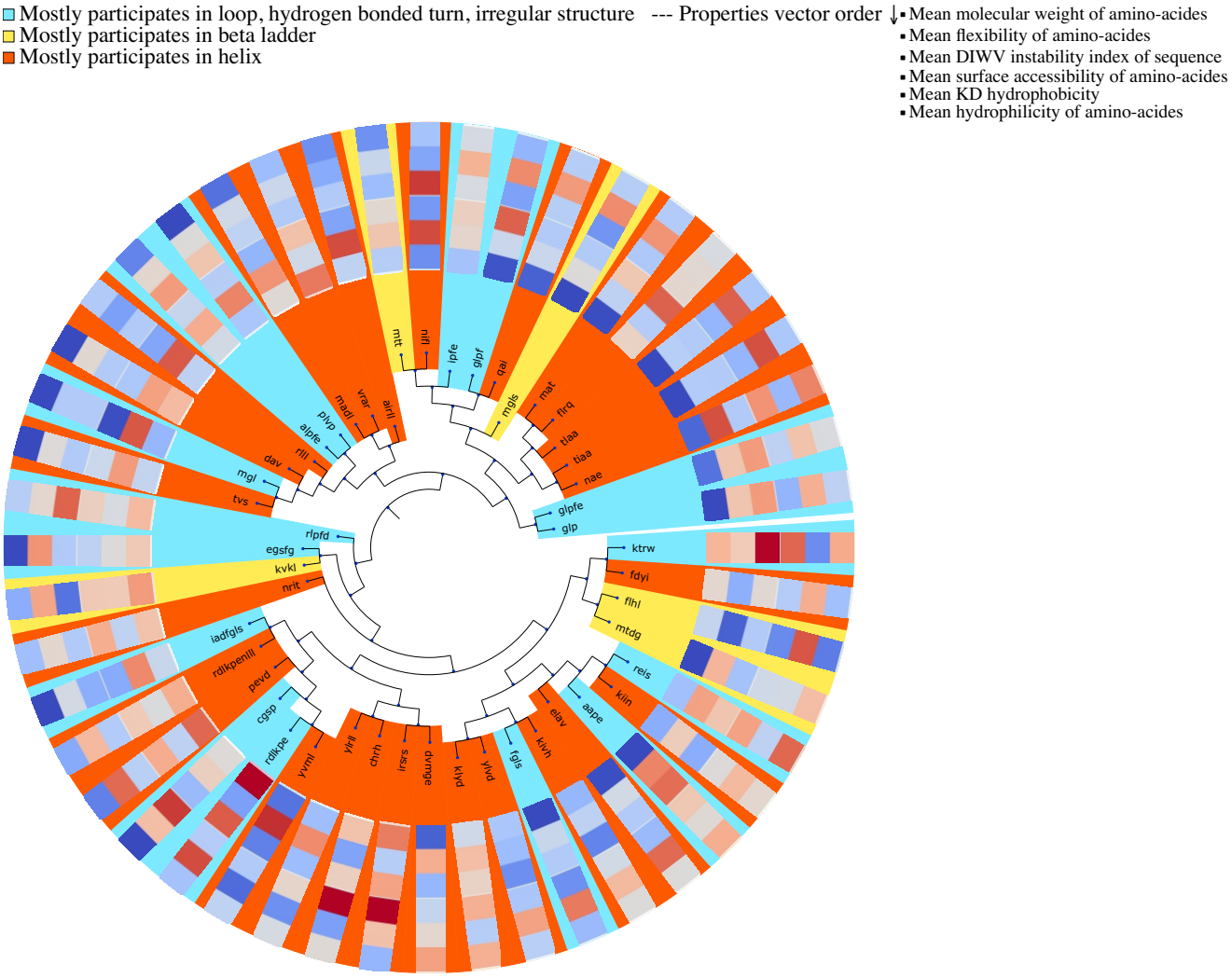

Figure 4. Clustering of biofilm formation-specific motifs based on their occurrence in the annotation proteins. Each motif is colored based on the most frequent secondary structure it assumes in the Protein Data Bank (PDB) structure out of all PDB sequences.

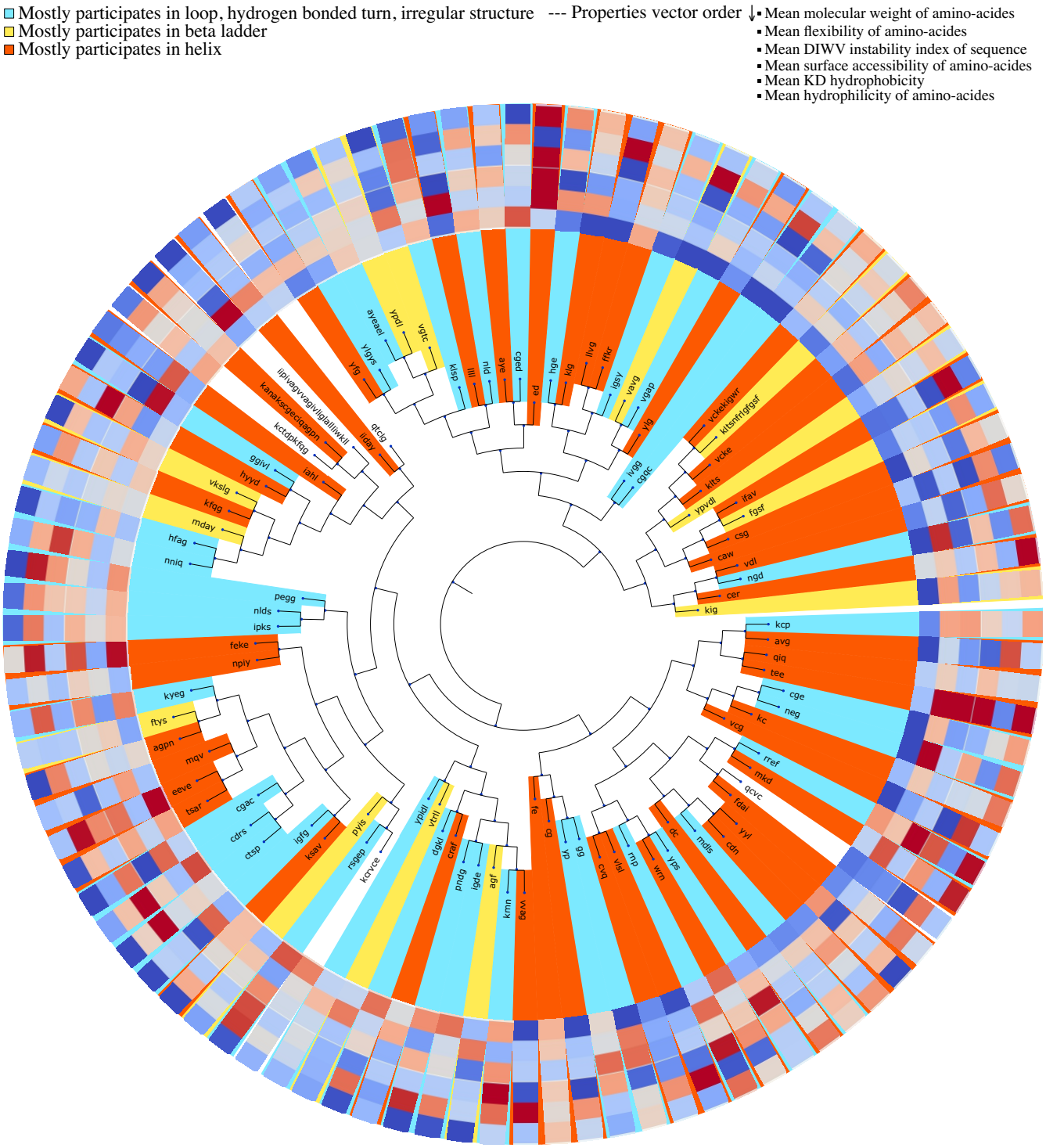

Figure 5. Clustering of integrin-related motifs based on their occurrence in the annotation proteins. Each motif is colored based on the most frequent secondary structure it assumes in the Protein Data Bank (PDB) structure out of all PDB sequences.

Table 1. Evaluation of protein sequence motifs mined with PPE motif mining for classification of integrin-binding and biofilm formation associated proteins. Support Vector Machine classifiers are tuned and evaluated in a stratified 10xfold cross-validation setting and then tested on a separate held-out dataset.

Dataset	Probabilistic Segmentation	Representation	10 fold cross-validation			Performance on the test set		
			Precision	Recall	F1	Precision	Recall	F1
Integrin-Binding	True	top-1000 (998 positive)	0.88	0.85	0.87	0.91	0.85	0.88
		top-100 (100 positive)	0.73	0.66	0.69	0.84	0.67	0.75
	False	top-1000 (982 positive)	0.91	0.87	0.89	0.93	0.86	**0.89**
		top-100 (100 positive)	0.73	0.68	0.70	0.84	0.67	0.75
Integrins	True	top-1000 (1000 positive)	0.94	0.76	0.84	1	0.75	0.86
		top-100 (100 positive)	0.91	0.82	0.86	0.83	0.83	**0.83**
	False	top-1000 (996 positive)	0.96	0.82	0.89	1	0.83	0.91
		top-100 (100 positive)	0.88	0.83	0.86	0.9	0.75	0.82
Biofilm formation	True	top-1000 (103 positive)	0.89	0.67	0.76	0.82	0.56	0.72
		top-500 (48 positive)	0.81	0.71	0.76	0.76	0.71	**0.73**
	False	top-1000 (53 positive)	0.79	0.67	0.73	0.74	0.66	0.70
		top-500 (26 positive)	0.78	0.67	0.72	0.73	0.65	0.69

Table 2. Evaluation of the significant NLS patterns against 3254 experimentally identified motifs.

PPE training dataset	Probabilistic Segmentation	Overlapping hits (> 3)	Hits more than 70% overlap	Number of exact matches
Swiss-Prot	True	3253	337	37
Swiss-Prot	False	3162	107	15
Nuclear Proteins	True	3253	381	42
Nuclear Proteins	False	3198	137	21

prediction.

4 Conclusions

We proposed a new unsupervised feature extraction from protein sequences. Instead of fixed-length k-mers, we segmented sequences into the most frequent variable-length sub-sequences inspired by Byte-pair encoding (BPE), a data compression algorithm. These sub-sequences were then used as data representation for downstream machine learning tasks. As a modification to the original BPE algorithm, we define a probabilistic segmentation by sampling from the space of possible vocabulary sizes. This allows for considering multiple views of segmenting a sequence into sub-sequences. The main purpose of this work has been introducing a variable-length segmentation of sequences, similar to word tokenization in natural languages. In particular we introduced (i) DiMotif as an alignment free discriminative protein sequence motif miner, as well as (ii) ProtVecX a variable-length extension of protein sequence embedding. We evaluated DiMotif in extracting motifs related to (i) integrins, (ii) integrin-binding proteins, (iii) biofilm formation. We showed that the extracted motifs could reliably detect held-out sequences of the same phenotypes by high F1 scores. We also show that DiMotif could reasonably detect experimentally identified motifs related to nuclear localization signal. Using KL-divergence between occurrence distribution of motifs in the positive sequences, DiMotif is capable of outputing multi-part motifs. We proposed ProtVecX embedding trained over sub-sequences over SwissProt. We demonstrated that concatenation of the raw k-mer distributions with the embedding representations can improve the sequence classification performances compared to use of only either of k-mers or embeddings. In addition, concatenation of ProtVecX to k-mers occurrences outperformed ProtVec embedding combined with k-mer occurrences in toxin and enzyme prediction tasks. The applications of this work is not limited to motif mining or embedding training and we expect this representation to be widely used in bioinformatics tasks.

Table 3. Comparing k-mers, ProtVec, and ProtVecX in protein classification tasks.

Dataset	Representation	5 fold cross-validation		
		macro-Precision	macro-Recall	macro-F1
Venom toxins prediction	3-mer	0.89	0.89	0.89
	ProtVec	0.88	0.88	0.88
	ProtVecX	0.88	0.88	0.88
	3-mer + ProtVec	0.90	0.89	0.89
	3-mer + ProtVecX	0.90	0.90	**0.90**
Subcellular location prediction	3-mer	0.65	0.59	0.60
	ProtVec	0.60	0.57	0.58
	ProtVecX	0.57	0.57	0.57
	3-mer + ProtVec	0.68	0.60	**0.62**
	3-mer + ProtVecX	0.66	0.60	0.61
Enzyme prediction	3-mer	0.70	0.73	0.71
	ProtVec	0.68	0.70	0.69
	ProtVecX	0.69	0.71	0.70
	3-mer + ProtVec	0.70	0.73	0.71
	3-mer + ProtVecX	0.71	0.73	**0.72**

Acknowledgements

Fruitful discussions with Hengameh Shams and Ardavan Saeedi are gratefully acknowledged.

References

1. Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, vol. 10, 707–710 (1966).
2. Waterman, M. S., Smith, T. F. & Beyer, W. A. Some biological sequence metrics. *Adv. Math.* **20**, 367–387 (1976).
3. Searls, D. B. The computational linguistics of biological sequences. *Artif. intelligence molecular biology* **2**, 47–120 (1993).
4. Searls, D. B. The language of genes. *Nat.* **420**, 211 (2002).
5. Asgari, E. & Mofrad, M. R. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS one* **10**, e0141287 (2015).
6. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K. & Dyer, C. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360* (2016).
7. Johansen, A. R., Sønderby, C. K., Sønderby, S. K. & Winther, O. Deep recurrent conditional random field network for protein secondary prediction. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 73–78 (ACM, 2017).
8. Yandell, M. D. & Majoros, W. H. Genomics and natural language processing. *Nat. Rev. Genet.* **3**, 601 (2002).
9. Grabherr, M. G. *et al.* Full-length transcriptome assembly from rna-seq data without a reference genome. *Nat. biotechnology* **29**, 644 (2011).
10. Jolma, A. *et al.* Dna-binding specificities of human transcription factors. *Cell* **152**, 327–339 (2013).
11. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nat. biotechnology* **33**, 831 (2015).
12. Wood, D. E. & Salzberg, S. L. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome biology* **15**, R46 (2014).
13. Asgari, E., Garakani, K., McHardy, A. C. & Mofrad, M. R. Micropheno: Predicting environments and host phenotypes from 16s rrna gene sequencing using a k-mer based representation of shallow sub-samples. *bioRxiv* 255018 (2018).
14. Asgari, E., Münch, P. C., Lesker, T. R., McHardy, A. C. & Mofrad, M. R. Nucleotide-pair encoding of 16s rrna sequences for host phenotype and biomarker detection. *bioRxiv* 334722 (2018).

15. Gage, P. A new algorithm for data compression. *The C Users J.* **12**, 23–38 (1994).

16. Chen, L., Lu, S. & Ram, J. Compressed pattern matching in dna sequences. In *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*, 62–68 (IEEE, 2004).

17. Sennrich, R., Haddow, B. & Birch, A. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909* (2015).

18. Frith, M. C., Saunders, N. F., Kobe, B. & Bailey, T. L. Discovering sequence motifs with arbitrary insertions and deletions. *PLoS computational biology* **4**, e1000071 (2008).

19. Edwards, R. J. & Palopoli, N. Computational prediction of short linear motifs from protein sequences. In *Computational Peptidology*, 89–141 (Springer, 2015).

20. Sobhy, H. A bioinformatics pipeline to search functional motifs within whole-proteome data: a case study of poxviruses. *Virus genes* **53**, 173–178 (2017).

21. Baichoo, S. & Ouzounis, C. A. Computational complexity of algorithms for sequence comparison, short-read assembly and genome alignment. *Biosyst.* **156**, 72–85 (2017).

22. Bailey, T. L. *et al.* Meme suite: tools for motif discovery and searching. *Nucleic acids research* **37**, W202–W208 (2009).

23. Palopoli, N., Lythgow, K. T. & Edwards, R. J. Qslimfinder: improved short linear motif prediction using specific query protein data. *Bioinforma.* **31**, 2284–2293 (2015).

24. Davey, N. E., Haslam, N. J., Shields, D. C. & Edwards, R. J. Slimsearch 2.0: biological context for short linear motifs in proteins. *Nucleic acids research* **39**, W56–W60 (2011).

25. Horn, H., Haslam, N. & Jensen, L. J. Doremi: context-based prioritization of linear motif matches. *PeerJ* **2**, e315 (2014).

26. Prytuliak, R., Volkmer, M., Meier, M. & Habermann, B. H. Hh-motif: de novo detection of short linear motifs in proteins by hidden markov model comparisons. *Nucleic acids research* gkx341 (2017).

27. Ba, A. N. N. *et al.* Proteome-wide discovery of evolutionary conserved sequences in disordered regions. *Sci. Signal.* **5**, rs1–rs1 (2012).

28. Liu, B., Yang, J., Li, Y., McDermaid, A. & Ma, Q. An algorithmic perspective of de novo cis-regulatory motif finding based on chip-seq data. *Briefings bioinformatics* bbx026 (2017).

29. Redhead, E. & Bailey, T. L. Discriminative motif discovery in dna and protein sequences using the deme algorithm. *BMC bioinformatics* **8**, 385 (2007).

30. Song, T. & Gu, H. Discriminative motif discovery via simulated evolution and random under-sampling. *PloS one* **9**, e87670 (2014).

31. Maaskola, J. & Rajewsky, N. Binding site discovery from nucleic acid sequences by discriminative learning of hidden markov models. *Nucleic acids research* **42**, 12995–13011 (2014).

32. Mehdi, A. M., Sehgal, M. S. B., Kobe, B., Bailey, T. L. & Bodén, M. Dlocalmotif: a discriminative approach for discovering local motifs in protein sequences. *Bioinforma.* **29**, 39–46 (2013).

33. Lanchantin, J., Singh, R., Wang, B. & Qi, Y. Deep motif dashboard: Visualizing and understanding genomic sequences using deep neural networks. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2017*, 254–265 (World Scientific, 2017).

34. Dinkel, H. *et al.* Elm—the database of eukaryotic linear motifs. *Nucleic acids research* **40**, D242–D251 (2011).

35. Bernhofer, M. *et al.* Nlsdb—major update for database of nuclear localization signals and nuclear export signals. *Nucleic acids research* **46**, D503–D508 (2017).

36. Collobert, R. *et al.* Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **12**, 2493–2537 (2011).

37. Tang, D. *et al.* Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 1555–1565 (2014).

38. Levy, O. & Goldberg, Y. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, 2177–2185 (2014).

39. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119 (2013).

40. Asgari, E. & Mofrad, M. R. Comparing fifty natural languages and twelve genetic languages using word embedding language divergence (weld) as a quantitative measure of language distance. In *In Proceedings of the NAACL-HLT Workshop on Multilingual and Cross-lingual Methods in NLP, San Diego, CA*, 65–74 (Association for Computational Linguistics, 2016).

41. Islam, S. A., Heil, B. J., Kearney, C. M. & Baker, E. J. Protein classification using modified n-grams and skip-grams. *Bioinforma.* btx823 (2017).

42. Min, S., Lee, B. & Yoon, S. Deep learning in bioinformatics. *Briefings bioinformatics* **18**, 851–869 (2017).

43. Kim, S., Lee, H., Kim, K. & Kang, J. Mut2vec: distributed representation of cancerous mutations. *BMC medical genomics* **11**, 33 (2018).

44. Jaeger, S., Fulle, S. & Turk, S. Mol2vec: Unsupervised machine learning approach with chemical intuition. *J. chemical information modeling* **58**, 27–35 (2018).

45. Du, J. *et al.* Gene2vec: Distributed representation of genes based on co-expression. *bioRxiv* 286096 (2018).

46. Consortium, U. Uniprot: the universal protein knowledgebase. *Nucleic acids research* **45**, D158–D169 (2016).

47. Emanuelsson, O., Brunak, S., Von Heijne, G. & Nielsen, H. Locating proteins in the cell using targetp, signalp and related tools. *Nat. protocols* **2**, 953 (2007).

48. Gacesa, R., Barlow, D. J. & Long, P. F. Machine learning can differentiate venom toxins from other proteins having non-toxic physiological functions. *PeerJ Comput. Sci.* **2**, e90 (2016).

49. Jungo, F. & Bairoch, A. Tox-prot, the toxin protein annotation program of the swiss-prot protein knowledgebase. *Toxicon* **45**, 293–301 (2005).

50. Li, Y. *et al.* Deepre: sequence-based enzyme ec number prediction by deep learning. *Bioinforma.* **1**, 10 (2017).

51. Boutet, E. *et al.* Uniprotkb/swiss-prot, the manually annotated section of the uniprot knowledgebase: how to use the entry view. In *Plant Bioinformatics*, 23–54 (Springer, 2016).

52. Kullback, S. & Leibler, R. A. On information and sufficiency. *The annals mathematical statistics* **22**, 79–86 (1951).

53. Vihinen, M., Torkkila, E. & Riikonen, P. Accuracy of protein flexibility predictions. *Proteins: Struct. Funct. Bioinforma.* **19**, 141–149 (1994).

54. Guruprasad, K., Reddy, B. B. & Pandit, M. W. Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence. *Protein Eng. Des. Sel.* **4**, 155–161 (1990).

55. Emini, E. A., Hughes, J. V., Perlow, D. & Boger, J. Induction of hepatitis a virus-neutralizing antibody by a virus-specific synthetic peptide. *J. virology* **55**, 836–839 (1985).

56. Kyte, J. & Doolittle, R. F. A simple method for displaying the hydropathic character of a protein. *J. molecular biology* **157**, 105–132 (1982).

57. Hopp, T. P. & Woods, K. R. Prediction of protein antigenic determinants from amino acid sequences. *Proc. Natl. Acad. Sci.* **78**, 3824–3828 (1981).

58. Rose, P. W. *et al.* The rcsb protein data bank: integrative view of protein, gene and 3d structural information. *Nucleic acids research* gkw1000 (2016).

59. Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* (2016).

60. Guan, J.-L. & Hynes, R. O. Lymphoid cells recognize an alternatively spliced segment of fibronectin via the integrin receptor $\alpha 4\beta 1$. *Cell* **60**, 53–61 (1990).

61. Ruoslahti, E. Rgd and other recognition sequences for integrins. *Annu. review cell developmental biology* **12**, 697–715 (1996).

62. Plow, E. F., Haas, T. A., Zhang, L., Loftus, J. & Smith, J. W. Ligand binding to integrins. *J. Biol. Chem.* **275**, 21785–21788 (2000).

63. Plow, E. F., Pierschbacher, M. D., Ruoslahti, E., Marguerie, G. A. & Ginsberg, M. H. The effect of arg-gly-asp-containing peptides on fibrinogen and von willebrand factor binding to platelets. *Proc. Natl. Acad. Sci.* **82**, 8057–8061 (1985).

64. Kapp, T. G. *et al.* A comprehensive evaluation of the activity and selectivity profile of ligands for rgd-binding integrins. *Sci. reports* **7**, 39805 (2017).

65. Ochsenhirt, S. E., Kokkoli, E., McCarthy, J. B. & Tirrell, M. Effect of rgd secondary structure and the synergy site phsrn on cell adhesion, spreading and specific integrin engagement. *Biomater.* **27**, 3863–3874 (2006).