MiPepid: MicroPeptide identification tool using machine learning

Mengmeng Zhu
Purdue University

Michael Gribskov (✉ mgiibsko@purdue.edu)
https://orcid.org/0000-0002-1718-0242

Software

Keywords: micropeptide, small ORF, sORF, smORF, coding, noncoding, IncRNA, machine learning

Posted Date: September 18th, 2019

DOI: https://doi.org/10.21203/rs.2.9710/v2

License: © This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Version of Record: A version of this preprint was published on November 8th, 2019. See the published version at https://doi.org/10.1186/s12859-019-3033-9.
Abstract

Background Micropeptides are small proteins with a length of <= 100 amino acids. Short open reading frames that could produce micropeptides were traditionally ignored due to technical difficulties, as few small peptides had been experimentally demonstrated. In the past decade, a growing number of micropeptides have been shown to play significant roles in vital biological activities. Despite the increased amount of data, we still lack bioinformatics tools for specifically identifying micropeptides from DNA sequences. Indeed, most existing tools for classifying coding and noncoding ORFs were built on datasets in which “normal-sized” proteins were considered to be positives and short ORFs were generally considered to be noncoding. Since the functional and biophysical constraints on small peptides are likely to be different from those on “normal” proteins, methods for predicting short translated ORFs must be trained independently from those for longer proteins.

Results In this study, we have developed MiPepid, a machine-learning tool specifically for the identification of micropeptides. We trained MiPepid using carefully cleaned data from existing databases and used logistic regression with 4-mer features. With only the sequence information of an ORF, MiPepid is able to predict whether it encodes a micropeptide with 96% accuracy on a blind dataset of high-confidence micropeptides, and to correctly classify newly discovered micropeptides not included in either the training or the blind test data. Compared with state-of-the-art coding potential prediction methods, MiPepid performs exceptionally well, as other methods incorrectly classify most bona fide micropeptides as noncoding. MiPepid is alignment-free and runs sufficiently fast for genome-scale analyses. It is easy to use and is available at https://github.com/MindAI/MiPepid.

Conclusions MiPepid was developed to specifically predict micropeptides, a category of proteins with increasing significance, from DNA sequences. It shows evident advantages over existing coding potential prediction methods on micropeptide identification. It is ready to use and runs fast.

keywords: micropeptide, small ORF, sORF, smORF, coding, noncoding, lncRNA, machine learning

Background

Micropeptides are generally defined as small proteins of <= 100 amino acid residues [1–3]. Their existence was traditionally ignored because few micropeptides had been shown to be functionally important, mostly due to technological limitations in isolating small proteins [4]. Consequently, small open reading frames (sORFs or smORFs, <= 303 bp) that encode micropeptides are generally ignored in gene annotation and are considered noise (occurring by chance) and to be unlikely to be translated into proteins [2, 4, 5].

With improved technology, an increasing number of micropeptides have been discovered, and have been shown to play important roles in muscle performance [6], calcium signaling [7], heart contraction [8], insulin regulation [9], immune surveillance [10, 11], etc. In particular, many micropeptides were shown to be translated from transcripts that were previously annotated as putative long noncoding RNAs (lncRNAs) [12, 13]. This fact challenges the “noncoding” definition and raises questions about the functional mechanisms of lncRNAs, i.e., whether they function through their 3D RNA structure, or via the micropeptides translated from encoded sORFs, or both.

With the increasing recognition of the importance of the “once well forgotten” field of micropeptides, it is increasingly important to develop a large-scale method for identifying them in a cost-effective way. Ribosome profiling [14, 15] (Ribo-Seq) is a recent high-throughput technique for identifying potentially coding sORFs by sequencing mRNA fragments captured with translating ribosomes. Despite its advantages, there currently is no
community consensus on how Ribo-Seq data should be used for gene annotation [16], as some investigators have questioned whether capture of RNAs by ribosomes necessarily implies translation; some capture could be transient or non-specific rather than truly functional [17, 18]. Ribo-Seq requires the use of next generation sequencing and thus has significant costs. In addition, depending on the sequencing depth and quality, it may suffer from false positives, and may not reveal all coding sORFs due to differences in sORF expression in different tissues, developmental stages, and conditions. Therefore, the sORFs discovered from Ribo-Seq still require experimental verification of their coding potentials.

It is much less expensive to predict coding sORFs from DNA sequences using bioinformatic tools. Although experimental verification is still required for predicted sORFs, a bioinformatic prediction of the coding potentials of any sORFs before experimental verification is valuable since bioinformatics analysis costs almost nothing and could potentially provide useful insights.

There are currently few bioinformatic tools specifically designed for predicting the coding potential of small ORFs. uPEPperoni [19] is a web server designed to detect sORFs in the 5’ untranslated regions (5’-UTR) of mRNAs. It detects conserved sORF without explicitly predicting coding potentials. Although 5’-UTR sORFs are an important component of the sORF population, many sORFs are located elsewhere, such as within the coding region of an mRNA, in IncRNAs, etc. sORF finder [20] is a program that specifically identifies sORFs using the nucleotide frequency conditional probabilities of the sequence, however it was developed nearly a decade ago (2009), and the server is no longer accessible. In addition, because many micropeptides have been discovered in the last decade, a much larger training dataset can now be assembled, and this should greatly improve the prediction quality. Data pipelines have been described [21–23] that calculate the coding abilities of sORFs, especially those identified from Ribo-Seq data; however, these pipelines are not standalone packages readily available for other users. Other well-known coding potential prediction tools such as CPC [24], CPC2 [25], CPAT [26], CNCI [27], PhyloCSF [28], etc. were trained on datasets consisting primarily of normal-sized proteins. Because of the differences between sORF peptides and globular proteins, and because these methods were not trained on large sORF datasets, it is likely they do not perform well in sORF prediction (as shown in the Results section below). In general, most coding potential predictors penalize short ORFs and those that lack significant similarity to known proteins; both of these factors compromise the ability of existing tools to correctly predict sORFs.

With the ongoing development of techniques such as Ribo-Seq and mass spectrometry (MS), an increasingly number of micropeptides have been experimentally identified and verified. We have a reasonable amount of data that can be leveraged for the development of bioinformatics tools specifically for micropeptide prediction. sORFs.org [4, 5] is a repository of small ORFs identified specifically from Ribo-Seq and MS data. And SmProt [29] is a database of micropeptides collected from literature mining, known databases, ribosome profiling, and MS.

Machine learning (ML) is a set of algorithms for learning hidden patterns within a set of data in order to classify, cluster, etc. The development of a successful ML-based method for a particular problem depends on a good dataset (clean, with sufficient data, etc.), and a good choice of specific ML algorithm. ML has been used in developing numerous bioinformatics tools, and has been used, for instance, in ORF coding potential prediction [24–27].
In this study, we developed MiPepid, a ML-based tool specifically for identifying micropeptides directly from DNA sequences. It was trained on a high-quality dataset carefully collected and cleaned by ourselves using the well-studied logistic regression model, and it achieves impressive performance on several blind test datasets. Compared with several existing state-of-the-art coding potential prediction tools, MiPepid performs exceptionally well on *bona fide* micropeptide datasets, indicating its superiority in identifying small-sized proteins. It is also a lightweight and alignment-free method that runs sufficiently fast and scales well.

**Methods**

**Datasets**

To collect a positive as well as a negative dataset for micropeptides that are representative yet concise, we selected 2 data sources: SmProt [29] and traditional noncoding RNAs.

**The positive dataset**

SmProt [29] is a database of small proteins / micropeptides which includes data from literature mining, known databases (UniProt [30], NCBI CCDS [31–33]), Ribo-Seq, and MS. In particular, SmProt contains a high-confidence dataset consisting of micropeptide data that were collected from low-throughput literature mining, known databases, and high-throughput literature mining data or Ribo-Seq data with supporting MS evidence.

The SmProt high-confidence dataset (containing 12,602 human micropeptides in total) is a reliable data source for positive data since many of the peptides have been experimentally verified, and the rest are supported by multiple evidence. Based on this dataset, we cleaned our own positive dataset using the following pipeline:

1. Obtain the nucleotide sequences of the data. In SmProt, only the amino acid sequences rather than the DNA sequences are provided, although for the majority of data points their corresponding transcript IDs (primarily in Ensembl [34], with others in RefSeq [34] or NONCODE [34]) are provided. Since the DNA sequence of a micropeptide contains essential information that the translated sequence cannot provide (such as nucleotide frequency, etc.), we therefore obtained the corresponding DNA sequences by mapping the protein sequences back to their corresponding transcripts using GeneWise [34]. To ensure the quality of the dataset, only micropeptides that gave a perfect match (no substitutions or indels) were retained.

2. Obtain a nonredundant positive dataset. Proteins with similar sequences may share similar functions, and families of related sequences create a bias towards certain sequence features. To ensure that our positive dataset is not biased by subgroups of micropeptides with similar sequences, we selected a nonredundant set with protein sequence identity ≤ 0.6. This serves as our positive dataset and it contains 4,017 data points.

**The negative dataset**

It is hard to define a truly negative dataset for micropeptides as more and more sequences that were formerly considered noncoding have been shown to encode translated proteins, such as 5’-UTRs of mRNAs, IncRNAs, etc.
Despite the limitations of our current knowledge, we are still able to collect ORFs that are highly likely to be noncoding.

Traditional noncoding RNAs, such as microRNA (miRNA), ribosomal RNA (rRNA), small nuclear RNA (snRNA), etc. are highly likely to be truly noncoding. While there is growing evidence that IncRNAs [35, 36] may sometimes encode translated sORFs, the possibility of sORFs in traditional noncoding RNAs has seldom been mentioned or discussed in literature. In addition, some pipelines for predicting coding regions from Ribo-Seq data utilized those ncRNAs to construct their negative datasets [21, 37]. While there are examples of IncRNAs and “noncoding” regions of mRNAs that encode micropeptides in the SmProt high-confidence dataset, there are no examples of micropeptides encoded by traditional ncRNAs.

We therefore chose human miRNA, rRNA, snRNA, snoRNA (small nucleolar RNA), tRNA (transfer RNA), and scaRNA (small Cajal body RNA, a nucleolar RNA) as the data source for our negative dataset. We selected all human transcripts in Ensembl database [34] with these 6 biotypes and extracted all possible ORFs from those transcripts, i.e., ORFs with valid start and stop codons from all 3 translation frames. Although there is evidence that non-ATG codons sometimes serve as sORF start codons [5], to ensure the validity of our dataset, we consider only ATG start codons in constructing the negative dataset; in the positive dataset, nearly 99% of sORFs begin with ATG start codons.

We finally gathered 5616 negative sORFs. In the same way as for the positive data, we selected a nonredundant negative dataset of size 2,936 with pairwise predicted protein sequence identity ≤ 0.6.

The training set and the blind test set

We randomly selected 80% of the examples in the positive and negative datasets to build our training set for machine learning model training; the remaining 20% were used as a blind test set which was only used for model evaluation (Table 1).

The synthetic_negative dataset

To further test the performance of our method, we generated a synthetic dataset that preserves the length distribution as well as the dinucleotide frequencies [38] of the negative dataset. Since this dataset mimics the negative data, our method is expected to predict negative on this dataset. This synthetic_negative dataset is of the same size as the negative dataset (2,936), and it was generated using the ushuffle software [39] in the MEME suite [40].

Methods

Feature generation

In machine learning, identifying a set of relevant features is the next important step toward constructing a classifier. A set of well-chosen features greatly facilitates differentiating between different classes.
In our study, we believe the key to determining whether a small ORF is translated lies in the nucleotide patterns in the sequence. A translated sORF should have a DNA sequence that is constrained by the physicochemical properties of the translated peptide, the preference of ribosome occupancy, the codon bias of the organism, etc.

-mer features have been widely used to effectively capture nucleotide patterns. A -mer is a subsequence of length , where is an integer ranging from 1 to as high as hundreds depending on the requirements of specific questions. For DNA -mers, there are only 4 types of nucleotides (A, T, C, and G), so the number of distinct -mers for a specific is. The -mer features are simply encoded as a vector of size (denoted as), with each value in the vector denoting the frequency of one unique -mer in the sequence. If we slide a window of length across the sequence from beginning to end with a step size of , we obtain -mers in total, where denotes the length of the sequence. Therefore, where is the norm of . To exclude the sequence length effects in, we can use the normalized -mer features, i.e., the fractional frequency of each -mer rather than the frequency itself. In this case, Regarding the choice of, a hexamer (i.e., 6-mer) is often used in bioinformatics tools for various biological questions [20, 41]. Yet hexamers would give a feature vector of size. Compared to , the size of our training data, a model with as many as parameters could potentially overfit the dataset although is larger than . To ensure the generalizability as well as the efficiency of our method, we chose to use 4-mer features. A 4-mer, while short, still captures information about codons, and any dependencies between adjacent amino acid residues since every 4-mer covers parts of 2 adjacent codons / amino acids. A 4-mer feature vector has a reasonable size of 256, much less than 4,096, therefore should produce less model overfitting and have shorter running time. To eliminate the length information of a sORF, we chose to use normalized k-mer features. And to better capture the codon information of the translation frame, we chose a step size of 3 for k-mer extraction.

**Logistic regression**

From many possible supervised machine learning algorithms, we chose logistic regression for our study. Logistic regression is well-studied and provides easy-to-interpret models that have been shown to be successful in numerous cases and scenarios. The model can be tuned to minimize overfitting by, for instance, including regularization penalties. When used for prediction, the model returns the probability of an instance being in the positive category rather than just a label, which gives more insight into the prediction.

The loss function for logistic regression is:

\[ L(w, b, y) = -\sum_{i=1}^{n} y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \]

where \( y_i \) are the set of the data points and for each \( i \), is the label. is the weight vector and is the bias term. is the negative log likelihood. is the regularization term which helps constrain the parameter space of to reduce overfitting, and is a hyperparameter controlling the regularization strength. For a set of and, the classifier assigns the label to data point based on the following:

\[ \hat{y}_i = \begin{cases} 1 & \text{if } p_i > \theta \text{ (positive class)} \\ 0 & \text{otherwise (negative class)} \end{cases} \]

where \( \theta \) is the predicted label from the classifier and is the threshold between the positive () and the negative () classes. Although is generally used, is also a tunable hyperparameter.

**Performance evaluation**

To evaluate the performance of MiPepid and existing methods, we used the following metrics.
(1) accuracy

For a dataset, denote the number of correctly classified cases by a method as \( \hat{y} \), then the accuracy is \( \frac{\sum_{i=1}^{n} \mathbb{1}(\hat{y}_i = y_i)}{n} \), where \( n \) is the size of the dataset. This definition applies to any dataset used in this paper.

(2) score

For a dataset that contains both positive and negative data, the score of the performance of a method on this dataset is:

\[
\text{score} = \frac{TP}{TP + FP} \times \frac{TP}{TP + FN}
\]

\[= \frac{TP}{TP + FP + FN} \]

where \( TP \) is the precision and \( FP \) is the recall, and

\[= \frac{TP}{TP + FP} \]

\[= \frac{TP}{TP + FN} \]

\[= \frac{TP}{TP + FP + FN} \]

\[= \frac{TP}{TP + FP + FN} \]

where \( TP \) is the number of true positives, \( FP \) is the number of false positives, \( FN \) is the number of false negatives, \( TP \) is the number of correctly classified cases in the positive subset; \( FP \) is the number of misclassified cases in the negative subset; \( FN \) is the number of false negatives, \( FN \) is the number of misclassified cases in the positive subset. The score ranges from 0 to 1, with a higher value implying better performance. In this study, the score is used for the training and the test sets, as both of them consist of both positive and negative data.

### 10-fold cross validation

\( N \)-fold cross validation is commonly used to select good hyperparameters. Here \( N \) is an integer ranging from 2 to as high as dozens. In cross validation, the dataset is randomly and evenly divided into folds. For every set of hyperparameter candidates, and for each fold, a model is trained using the other fold(s) and is evaluated on the left-out fold. The (weighted) average of the evaluations is taken as the overall evaluation for that set of hyperparameter candidates. This cross validation is done for every set of hyperparameter candidates in order to select a set that gives the best performance.

As stated in 3.3, there are 2 hyperparameters in logistic regression: the regularization strength and the threshold. We performed 10-fold cross validation to tune these 2 hyperparameters. For and , we selected the combination of and that gave the best performance.

### Results

#### Results on tuning hyperparameters using 10-fold cross validation

As stated in the “Methods” section, in the logistic regression model, both the regularization strength and the threshold are tunable hyperparameters. Therefore, before training the model on the training dataset, we firstly determined the best combination of and using 10-fold cross validation. As shown in Figure 1, when and, both the average (0.9639) and accuracy (0.9585) on the 10 validation sets are the highest.

**Fig. 1** 10-fold cross validation results with different \( \lambda \) and \( t \) combinations on the training set. : the hyperparameter for regularization strength in logistic regression; : the hyperparameter for threshold in logistic regression; best : when is fixed, the from that gives the best performance; avg val: the average score on the 10
validation sets when both and are fixed; avg accu val: the average accuracy on the 10 validation sets when both and are fixed.

**Results on the training set using the tuned hyperparameters**

We therefore chose and, and re-trained on the complete training dataset to obtain the MiPepid model. This model achieved an score of 0.9845 and an overall accuracy of 0.9822 on the training set (Table 2).

**MiPepid generalized well on the hold-out blind test set**

The blind test set contains 1,390 sequences and was not used during the training stage. As shown in table 3, MiPepid achieved an score of 0.9640 and an overall accuracy of 0.9576 on this test set. Compared with table 2, although the results are slightly lower, they are still comparably good. In addition, MiPepid performed almost equally well on the positive and negative subsets of the test set as indicated by the corresponding accuracies (0.9587 vs. 09559). Therefore, MiPepid generalizes well and has a balanced performance on both positive and negative data.

**MiPepid performed well on the synthetic_negative dataset**

The synthetic_negative dataset mimics the negative dataset by preserving the dinucleotide frequency as well as the length distribution of the real negative data, but because it has been randomized, should have no true sORFs. MiPepid achieved an accuracy of 0.9659 on the synthetic_negative dataset, a very close result to the one on the negative subset of either the training or test set, indicating the robustness of MiPepid.

**MiPepid correctly classifies newly published micropeptides**

In the positive dataset, part of the data were collected by low-throughput literature mining in SmProt [29], i.e., they were biologically/ experimentally verified on the level of protein, cell, phenotype, etc. SmProt [29], which was released in 2016, is based on literature published by Dec 2015. We searched for new examples of verified micropeptides, supported by extensive experimental evidence, published after Dec 2015, and found 5 new micropeptides in the literature (Table 4). Among these 5 cases, 3 are actually already recorded in SmProt [34], however they were in the non-high-confidence subset, i.e., there was only indirect evidence on the presence of those micropeptides.

These 5 cases were taken as the new_positive dataset. They are analogous to “the future cases” if the time boundary were Dec 2015. One of the major purposes of MiPepid is for future prediction. Therefore, its performance on “future cases” matters.

We applied MiPepid on this new_positive dataset, and MiPepid correctly classified all of the 5 micropeptides. And this is another result showing the good generalization of MiPepid.
Comparison with existing methods

Comparison with current ORF coding potential prediction methods

There are several state-of-the-art bioinformatics methods built to predict the coding/noncoding capability of a DNA sequence, including CPC [24], CPC2 [25], CPAT [26], CNIT [27], PhyloCSF [28], etc. However, all of them were designed to work on “average” transcript datasets, i.e., datasets that consist primarily of transcripts of regular-sized proteins and noncoding RNAs. In these methods, sORFs present in either an mRNA encoding a regular protein, or in a noncoding RNA, are generally penalized and are likely to be classified as noncoding; in the former case there is already a longer ORF present so shorter ones are treated as noncoding, and in the latter case the ORFs are automatically considered to be noncoding because they are found in “noncoding” RNAs. Therefore, despite the good performance of these methods in predicting regular-sized proteins, they may not be able to identify micropeptides, which also play critical biological roles.

In contrast, MiPepid is specifically designed to classify small ORFs in order to identify micropeptides. Here we chose CPC [24], CPC2 [25], and CPAT [26] as representatives of current methods and evaluated their performances on the hold-out blind test set as well as on the new_positive dataset, both of which the positive data are composed of high-confidence micropeptides.

As shown in table 5, while the 3 methods (CPC [24], CPC2 [25], CPAT [26]) performed exceptionally well on negative cases (100% accuracy), they indeed struggled to classify the positive cases.

The positive cases in the blind test set are sORFs of high-confidence micropeptides supported by at least 2 different types of experimental evidence. CPC [24] and CPC2 [25] considered over 90% of them as noncoding, while CPAT [26] did better with 32% accuracy but is still below half. In contrast, while MiPepid performed slightly worse on the negative cases (96%), it correctly classified 96% of the high confidence micropeptides. And regarding sORFs of the newly-published micropeptides, all of which are supported by protein-level and phenotypic evidence, CPC [24] and CPC2 [25] did not consider any of them to be coding, and CPAT [26] correctly classified only 3 out of 5. These results are not surprising as all three existing methods were trained on datasets primarily consisting of regular-sized proteins. It is clear from those results that sORFs are a special subpopulation of ORFs and predictions on which entail specially designed methods.

Comparison with sORFfinder

As mentioned in the Introduction section, sORFfinder predicts sORFs by calculating nucleotide frequency conditional probabilities of hexamers; however, the server is no longer accessible. We located a downloadable version at http://hanadb01.bio.kyutech.ac.jp/sORFfinder/ and ran it locally. sORFfinder does not provide a trained model for human sORFs, nor is there any human dataset included in this software. To conduct the comparison, we therefore used sORFfinder to train a model using our own training dataset and then evaluated on our test set. It took hours to train the model using sORFfinder, as compared to seconds when we trained.

As shown in table 6, sORFfinder was able to correctly predict around 87% examples in the test set, which is fairly good. However, it is clear that MiPepid performs significantly better. It is not surprising that sORFfinder achieved
a close performance to that of MiPepid. sORFfinder utilizes hexamer information and a naïve Bayes approach to calculate the posterior coding probability of a sORF given its hexamer composition. MiPepid uses 4-mer information, but rather than naïve Bayes, uses logistic regression to learn patterns from the data automatically. Notably, MiPepid achieves better classification using a much smaller feature vector, and much less computational time for training the model.

**Discussion**

**MiPepid’s prediction on the non-high-confidence subset of SmProt database**

The SmProt database has a high-confidence subset, examples of micropeptides that are supported by multiple kinds of evidence; the rest data are non-high-confidence. We collected those data and obtained their corresponding DNA sequences using the same pipeline used for the positive dataset (see Methods). We then used MiPepid to predict the coding capabilities of those data. Overall, MiPepid predicted 74% of them as positive. Table 7 shows detailed results based on different data sources.

As can be seen in table 7, among the over 25k sORFs collected by high-throughput literature mining, MiPepid predicted 80% of them as positive, which is a fairly high proportion. There are only 324 sORFs derived from MS data, and MiPepid labeled 72% of them as positive. Note that, on average, MS sORFs are significantly shorter than those from other sources. In contrast, among the over 13k Ribo-Seq derived sORFs, MiPepid only predicted 63% of them as positive. This is not very surprising as there has been debate on the credibility of Ribo-Seq data; some investigators have argued that the capture of an RNA transcript by the ribosome does not always lead to translation [16], and that some of the ribosome associated RNAs found in Ribo-Seq may be regulatory or non-specifically associated.

We are interested in looking at the relationship between the length of a sORF and MiPepid’s predicted coding probability.

**Fig. 2** Scatterplot of the length of sORF vs. predicted coding probability for the non-high-confidence sORFs in SmProt. aa: number of amino acids. The horizontal line separates sORFs that are predicted as positive (predicted coding probability ) and the rest predicted negative.

Figure 2 shows a moderately positive trend between the length of a sORF and MiPepid's predicted coding probability. This is reasonable considering the following: (1) the longer a sORF, the less likely it occurs by chance; (2) the longer a sORF, the more 4-mer information it contains, which helps MiPepid to better classify it. Yet, we do see that for many very short sORFs (< 20 aa), MiPepid was able to identify the positives, and for long sORFs (> 50 aa), MiPepid was not misled by the length, and was still able to identify some as negatives. In figure 7, one can also see that sORFs derived from the MS data are very short (< 30 aa).

**MiPepid’s prediction on uORFs of protein-coding transcripts**
A uORF (upstream open reading frame) is an ORF (usually short) located in the 5'-UTR (untranslated region) of a protein-coding transcript. A number of uORFs have been discovered to encode micropeptides and to play important roles in biological activities [47], and Ribo-Seq evidence suggested many uORFs are translated [19]. uORFs have been drawing increasing attention, and there is of great interest in determining the coding potentials of uORFs.

We extracted all possible small uORFs (from all 3 translation frames) of all annotated protein-coding transcripts in the Ensembl [34] human database. We then used MiPepid to determine the coding potentials of the extracted uORFs.

From 12,221 protein-coding transcripts, we extracted 42,589 small uORFs in total. 34.24% of the uORFs were predicted by MiPepid as coding. Among the 12,221 transcripts, 55.80% of them (6820) contain at least one potential micropeptide-encoding uORF. For the readers’ interest, we compiled all the small uORFs together with their coding potential score, location in the corresponding transcript, etc. into a file. This file is available along with the MiPepid package.

**MiPepid’s prediction on IncRNAs**

Long noncoding RNAs (IncRNAs) are RNA transcripts that lack a long ORF, therefore were initially considered untranslated. Yet a growing number of IncRNAs have been discovered to be actually translated into functional micropeptides [36, 43, 45, 48].

We extracted all possible sORFs (from all 3 translation frames) of all human IncRNA transcripts in Ensembl [34] (with biotype non_coding, 3prime_overlapping_ncRNA, antisense, lincRNA, retained_intron, sense_intronic, sense_overlapping, macro_IncRNA, or bidirectional_promoter_IncRNA). From the 26,711 IncRNA transcripts, we extracted 371,123 sORFs, averaging ~ 14 sORFs per transcript. 31.28% of the sORFs were predicted as coding. 86.63% of IncRNA transcripts were predicted to have at least one sORF that could potentially be translated into a micropeptide.

We present MiPepid’s prediction results on IncRNAs not for evaluating its performance but for showing that the proportion of sORFs in IncRNAs that are “similar” to sORFs of high-confidence micropeptides in our training set is very high. It is impossible to evaluate MiPepid using the IncRNA results as we have very little data on which sORFs in IncRNAs are truly positive, and which are not. The results serve as a reference for researchers interested in further work on any of those IncRNAs. The file for MiPepid’s results on the 26,711 annotated IncRNAs also available in the MiPepid software package.

**Conclusion**

MiPepid is designed to take a DNA sequence of a sORF and predict its micropeptide-coding capability. We suggest using sequences with transcriptome-level evidence, i.e., DNA sequences that are indeed transcribed, as MiPepid was trained to determine whether a transcript can be translated, and the training data did not include sORFs from untranslated DNA regions. The potential for an untranslated DNA sequence, such as an intergenic region, to be transcribed and translated was not addressed. MiPepid was specifically developed to predict small ORFs and “regular-sized” ORFs were not included in the training. Therefore, we recommend using MiPepid only
on sORFs; MiPepid is not trained to efficiently predict long ORFs such as those found in typical mRNAs. MiPepid was trained on human data, but should work for related mammalian species, such as mouse, rat, etc. Retraining the model on other species requires only a set of known micropeptides and the corresponding genomic sequence.

**List Of Abbreviations**

sORF / smORF: small open reading frame

IncRNA: long noncoding RNA

Ribo-Seq: Ribosome profiling

5'-UTR: 5' untranslated region

MS: mass spectrometry

ML: machine learning

miRNA: microRNA

rRNA: ribosomal RNA

snRNA: small nuclear RNA

snoRNA: small nucleolar RNA

tRNA: transfer RNA

scaRNA: small Cajal body RNA

uORF: upstream open reading frame

**Declarations**

**Availability of data and material**

The MiPepid software is available at: [https://github.com/MindAI/MiPepid](https://github.com/MindAI/MiPepid).

**Authors' contributions**

MZ developed the original concept, wrote all the code, performed all the experiments, and wrote the manuscript. MG and MZ regularly discussed details of the project, including which datasets to collect, which algorithms to use, how to improve performance, etc., and MG helped revise and proofread the manuscript. All authors have read and approved this manuscript.
Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

Not applicable.

Funding

Not applicable.

Acknowledgements

Not applicable.

References


Tables

Table 1. The sizes of training and test sets

<table>
<thead>
<tr>
<th>dataset</th>
<th>#positive1</th>
<th>#negative2</th>
<th>#total3</th>
</tr>
</thead>
<tbody>
<tr>
<td>training</td>
<td>3,194</td>
<td>2,369</td>
<td>5,563</td>
</tr>
<tr>
<td>test</td>
<td>823</td>
<td>567</td>
<td>1390</td>
</tr>
</tbody>
</table>

1 number of positive data points
2 number of negative data points
3 total number of data points

Table 2. MiPepid results on the training set

<table>
<thead>
<tr>
<th>F1</th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>positive</td>
</tr>
<tr>
<td>0.9845</td>
<td>0.9818</td>
</tr>
</tbody>
</table>

“positive” and “negative” refer to the accuracies of MiPepid on the positive and negative subsets, respectively; “overall” refers to the accuracy on the whole training set (positive + negative).

Table 3. MiPepid results on the blind test set

<table>
<thead>
<tr>
<th>F1</th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>positive</td>
</tr>
<tr>
<td>0.9640</td>
<td>0.9587</td>
</tr>
</tbody>
</table>

“positive” and “negative” refer to the accuracies of MiPepid on the positive and negative subsets, respectively; “overall” refers to the accuracy on the whole test set (positive + negative).

Table 4. List of micropeptides published after Dec 2015
<table>
<thead>
<tr>
<th>micropeptide name</th>
<th>protein sequence length</th>
<th>in SmProt non-highConf</th>
<th>reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOXI</td>
<td>56</td>
<td>yes</td>
<td>[42]</td>
</tr>
<tr>
<td>DWORF</td>
<td>35</td>
<td>yes</td>
<td>[43]</td>
</tr>
<tr>
<td>Myomixer / Minion</td>
<td>84</td>
<td>yes</td>
<td>[44]</td>
</tr>
<tr>
<td>SPAR</td>
<td>90</td>
<td>no</td>
<td>[45]</td>
</tr>
<tr>
<td>HOXB-AS3</td>
<td>53</td>
<td>no</td>
<td>[46]</td>
</tr>
</tbody>
</table>

in SmProt non-highConf: If this micropeptide was already included in the SmProt [34] non-high-confidence subset, then the value is "yes", otherwise "no".

Table 5. Comparison with existing methods on the blind test set and the new_positive dataset

<table>
<thead>
<tr>
<th>method</th>
<th>blind test set</th>
<th>new_positive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>positive</td>
<td>negative</td>
</tr>
<tr>
<td></td>
<td>#correct</td>
<td>accuracy</td>
</tr>
<tr>
<td>CPC [24]</td>
<td>17</td>
<td>0.02</td>
</tr>
<tr>
<td>CPC2 [25]</td>
<td>61</td>
<td>0.07</td>
</tr>
<tr>
<td>CPAT [26]</td>
<td>261</td>
<td>0.32</td>
</tr>
<tr>
<td>MiPepid (our method)</td>
<td>789</td>
<td>0.96</td>
</tr>
</tbody>
</table>

positive: the positive subset of the blind test set;

negative: the negative subset of the blind test set;

overall: the overall performance on the blind test set;

#correct: the number of correctly classified cases by a method;

accuracy: #correct divided by the total number of cases in that dataset/subset;

Table 6. Comparison with sORFfinder
<table>
<thead>
<tr>
<th>method</th>
<th>blind test set</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>positive</td>
<td>negative</td>
<td>overall</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>#correct</td>
<td>accuracy</td>
<td>#correct</td>
<td>accuracy</td>
<td>F1</td>
<td>accuracy</td>
<td></td>
</tr>
<tr>
<td>sORFfinder</td>
<td>708</td>
<td>0.86</td>
<td>506</td>
<td>0.89</td>
<td>0.89</td>
<td>0.87</td>
<td></td>
</tr>
<tr>
<td>MiPepid (our method)</td>
<td>789</td>
<td>0.96</td>
<td>542</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td></td>
</tr>
</tbody>
</table>

positive: the positive subset of the blind test set;

negative: the negative subset of the blind test set;

overall: the overall performance on the blind test set;

#correct: the number of correctly classified cases by a method;

accuracy: #correct divided by the total number of cases in that dataset/subset;

F1: the F1 score

Table 7. MiPepid’s prediction on the non-high-confidence data in SmProt

<table>
<thead>
<tr>
<th>data source</th>
<th>#sORFs</th>
<th>avg sORF length (aa)</th>
<th>#predicted positive</th>
<th>proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>high-throughput literature mining</td>
<td>25663</td>
<td>44</td>
<td>20516</td>
<td>0.80</td>
</tr>
<tr>
<td>ribosome profiling</td>
<td>13715</td>
<td>36</td>
<td>8596</td>
<td>0.63</td>
</tr>
<tr>
<td>MS data</td>
<td>324</td>
<td>15</td>
<td>233</td>
<td>0.72</td>
</tr>
</tbody>
</table>

high-throughput literature mining: published sORFs that were identified using high-throughput experimental methods;

ribosome profiling: sORFs predicted from Ribo-Seq data;

MS data: sORFs predicted from MS data;

#sORFs: number of sORFs from a particular data source;

avg sORF length (aa): the average length of sORFs measured in number of amino acids;

#predicted positive: number of sORFs that are predicted as positive by MiPepid;

proportion:

Figures
Figure 1

Parameter Optimization. The avg F1 and accuracy are shown at the best t for the indicated values of $\lambda$.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$10^{-5}$</th>
<th>$10^{-4}$</th>
<th>$10^{-3}$</th>
<th>$10^{-2}$</th>
<th>$10^{-1}$</th>
<th>$10^{0}$</th>
<th>$10^{1}$</th>
<th>$10^{2}$</th>
<th>$10^{3}$</th>
<th>$10^{4}$</th>
<th>$10^{5}$</th>
<th>$10^{6}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>best t</td>
<td>.60</td>
<td>.60</td>
<td>.60</td>
<td>.60</td>
<td>.65</td>
<td>.60</td>
<td>.60</td>
<td>.55</td>
<td>.50</td>
<td>.50</td>
<td>.50</td>
<td>.50</td>
</tr>
<tr>
<td>avg F1 val</td>
<td>.9635</td>
<td><strong>.9639</strong></td>
<td>.9625</td>
<td>.9507</td>
<td>.9450</td>
<td>.9414</td>
<td>.9364</td>
<td>.8088</td>
<td>.7292</td>
<td>.7292</td>
<td>.7292</td>
<td>.7292</td>
</tr>
<tr>
<td>avg accu val</td>
<td>.9581</td>
<td><strong>.9585</strong></td>
<td>.9567</td>
<td>.9428</td>
<td>.9365</td>
<td>.9306</td>
<td>.9261</td>
<td>.7293</td>
<td>.5741</td>
<td>.5741</td>
<td>.5741</td>
<td>.5741</td>
</tr>
</tbody>
</table>

Figure 2

Predicted Coding Probability. Coding probability as a function of the predicted small ORF length.