

Peer Review Information

Journal: Nature Computational Science

Manuscript Title: Unbiased organism-agnostic and highly sensitive signal peptide predictor with deep protein language model

Corresponding author name(s): Dr Yu Li

Editorial Notes:

Reviewer Comments & Decisions:

Decision Letter, initial version:
--

Date: 4th October 23 16:20:55

Last Sent: 4th October 23 16:20:55

Triggered By: Fernando Chirigati

From: fernando.chirigati@us.nature.com

To: liyu@cse.cuhk.edu.hk

CC: jie.pan@us.nature.com

BCC: fernando.chirigati@us.nature.com

Subject: Decision on Nature Computational Science manuscript NATCOMPUTSCI-23-0326A

Message: ** Please ensure you delete the link to your author homepage in this e-mail if you wish to forward it to your co-authors. **

Dear Dr Li,

Your manuscript "USPNet: unbiased organism-agnostic and highly sensitive signal peptide predictor with deep protein language model" has now been seen by 4 referees, whose comments are appended below. You will see that while they find your work of interest, they have raised points that need to be addressed before we can make a decision on publication.

The referees' reports seem to be quite clear. Naturally, we will need you to address *all* of the points raised.

While we ask you to address all of the points raised, the following points need to be substantially worked on:

- To the best of your abilities, please improve the writing of the paper and the presentation of the materials, so that the contributions of the method and the description of the results are clearer.
- Results section needs improvement; in particular:
 - * Not all of the methods seem to be depicted in the figures.
 - * A full evaluation adopting a complete set of scores should be reported.
 - * It is not clear how the datasets were used to train and evaluate the method. In particular, the benchmark and training datasets have a large overlap.
 - * Methods should also be compared by excluding proteins with unsatisfied MSA quality.
 - * The paper needs some ablation studies so that readers can better understand the novel technical contributions of the work.
- Please replace all radar plots and pie charts for more informative plots.

Please use the following link to submit your revised manuscript and a point-by-point response to the referees' comments (which should be in a separate document to any cover letter):

[REDACTED]

** This url links to your confidential homepage and associated information about manuscripts you may have submitted or be reviewing for us. If you wish to forward this e-mail to co-authors, please delete this link to your homepage first. **

To aid in the review process, we would appreciate it if you could also provide a copy of your manuscript files that indicates your revisions by making use of Track Changes or similar mark-up tools. Please also ensure that all correspondence is marked with your Nature Computational Science reference number in the subject line.

In addition, please make sure to upload a Word Document or LaTeX version of your text, to assist us in the editorial stage.

If you have any issues when updating your Code Ocean capsule during the revision process, please email the Code Ocean support team Cc'ing me.

To improve transparency in authorship, we request that all authors identified as 'corresponding author' on published papers create and link their Open Researcher and Contributor Identifier (ORCID) with their account on the Manuscript Tracking System (MTS), prior to acceptance. ORCID helps the scientific community achieve unambiguous attribution of all scholarly contributions. You can create and link your ORCID from the home page of the MTS by clicking on 'Modify my Springer Nature account'. For more information please visit www.springernature.com/orcid.

We hope to receive your revised paper within three weeks. If you cannot send it within this time, please let us know.

We look forward to hearing from you soon.

Best,

Fernando (on behalf of Jie Pan)

--

Fernando Chirigati, PhD
Chief Editor, Nature Computational Science
Nature Portfolio

Reviewers comments:

Reviewer #1 (Remarks to the Author):

The paper describes a new tool for the prediction of signal peptides.

Basically, the main novelties reside in

1 - the ability to perform prediction without indicating the source organism, that make the method useful for metagenomic studies. In this respect, also SignalP6 is organism-agnostic (as correctly commented in the paper). USPSignal is better performing, although the increment is not so astonishing (as expected, since prediction performance of the state-of-the-art is already very high)

2 - the adoption of a balancing procedure to improve the predictive performance. LDAM loss function has been adapted from a previous, still arXived paper, and seems effective.

The writing of the paper and the presentation of the material is often unclear and unfocused and this prevents the complete understanding.

Result presentation is somehow confusing, different plots are presented in figs 2-4 and possibly they should be better organized, improving readability and better defining the flux of information.

authors claim to compare 16 methods, but few of them are actually represented in fig 2 (panels a,b,d, and e - legends include more methods than reported). I agree that MCC and MCC2 are important indexes, but a full evaluation adopting a complete set of scores should be reported in supplementary materials. Also, color scale are difficult to be read and numbers should be provided.

Benchmark dataset seems to include proteins sharing a large sequence identity with the training set (Fig 2b, up to 90%).

Fair evaluation and comparison requires either to reduce the level of similarity or to rely on the independent testing set (whose evaluation and comparative assessment reported in fig 4 is quite limited)

The evaluation of the fast method in relation to the MSA based one is reported only for discrimination (fig 2d) and not for cleavage site prediction. Still, fig 2d is unclear and it is difficult to assess the effective performance of the method.

Authors introduce the possibility for the user to "adjust weights according to their specific needs, enhancing USPNet's effectiveness and suitability for their application scenarios." This is mostly unclear: on which basis a user can change parameters? and how, operatively?

A few questions are worth an investigation (not essential):

- Models adopted for encoding generally depends on the whole sequence, bidirectionally. What is the effect of truncating the sequence at 70 residues (pre-encoding) instead of truncating the encoding resulting from the whole sequence?
- In metagenomic studies sometimes assembled contigs show truncations at the N-ter. How robust is the method in this case?

Minor:

spell check is needed: just two examples from the first sentences in the introduction
 Secretary --> secretory
 Von Heijen --> Von Heijne

Hidden Markov models can be also used as supervised models

Reviewer #2 (Remarks to the Author):

MAJOR REVISIONS:

After reading the whole paper, it is not clear to me how the datasets were used to train and evaluate the method.

In the Methods section, three datasets are detailed, namely: i) training, ii) benchmark test and iii) independent test.

In line 186, the authors report that "Besides the benchmark set, we also conduct 5-fold cross-validation on the full training and benchmark sets (Supplementary Table 5-9). Since homology partitioning is inoperative in cross-validation, USPNet showcased more impressive performance, especially for the minor classes such as Tat/SPII and Sec/SPIII signal peptides."

Moreover, from Figure 2B, it seems like the benchmark dataset is not homology reduced against the training dataset as there are sequences with up to 90% similarity.

My understanding is that the benchmark dataset was adopted to select the model architecture and hyperparameters in a cross-validation setting, as well as for reporting results in Figure 2 to benchmark against other methods. The independent test dataset is instead only adopted to report results in section 2.5.

If this is the case, it would be better to report only comparative results against other methods on the completely independent blind test set and leave the so-called benchmark dataset as a validation dataset adopted to select the model. If my understanding is not correct, it would be better to rephrase the paper to better explain the training and validation procedures

Regarding USPNet-fast, the authors should try ProtT5 and ESM2 instead of ESM-1b, which generally performs worse, and they could provide comparisons between the different language models.

MINOR REVISIONS:

Figure 1B is confusing as the arrows go in different directions and it is not easy to follow the pipeline of the method
Figure 2D, radar plots are not Very informative

Reviewer #2 (Remarks on code availability):

I went to what is in the link and this is correct. I did not install the code for lack of time and for difficulties with my laptop.

Reviewer #3 (Remarks to the Author):

The authors have developed a Bi-LSTM-based method, USPNet, for predicting the types of signal peptides. USPNet incorporates protein language model and specially designed loss function to resolve the group information dependency and data imbalance problem in signal peptide prediction. Authors then build a pipeline and try to use genome data to discover novel signal peptides. They finally provide 347 sequences that are likely to be novel SPs. Some of the SPs have been supported by literature.

The comprehensive experiments prove USPNet has good generalization, and USPNet has good performance even if the sequence similarity is low. It is effective in domain-shift data as well as proteome-wide study. The methodology is decent to be applied outside of the SP field since data imbalance problem is common in protein-related tasks. The main contribution of this work is the authors develop a pipeline to discover potentially novel signal peptides from porcine gut, it started from the genome data and ends up with the signal peptide candidates. The pipeline is handy and can produce plenty of candidate sequences at a reasonable time. It is quite interesting that although the model just takes amino acid sequences as input and does not directly include protein tertiary structure information during the training stage, it screens the candidate sequences mainly based on the structural similarity instead of the sequence similarity. And the literature-based evaluation provides evidence for their experimental conclusions.

The manuscript is written in a clear and understandable manner, and the results are consistent with authors' conclusions. I have a few suggestions to improve the quality of the manuscript.

1) In Figure 2.e the authors mentioned that the incorrect predictions normally have poor Neff scores. If possible, authors might compare USPNet with USPNet-fast by excluding these proteins with unsatisfied MSA quality.

2) There is a lack of Precision and Recall of USPNet-fast in Figure 2.e. The result should be provided to show the effectiveness of USPNet-fast in cleavage site prediction.

3) Line 94, 'discovered' -> 'discover'.

4) In Figure 3.a, for the MSA and ESM-1b embeddings, are they obtained from the

original pre-trained model or from the model fine-tuned on the signal peptide training data? Authors might provide both for comparison to show if there is any change after fine-tuning.

5) The authors miss some spaces between text and citations/brackets.

6) Figure 5.a, a typo: 'Swine gut meragenome collection'

Reviewer #3 (Remarks on code availability):

Reproducibility of Results: users can reproduce the results presented in the paper using the provided code and datasets. The results were largely consistent with those described in the manuscript with minor variations that can be attributed to stochastic elements of the model. The differences were within acceptable margins, implying that the research is indeed replicable.

The code repository includes a comprehensive README file that provides clear instructions on the prerequisites, installation process, and steps to execute the code.

All necessary dependencies are listed. Authors have provided an environment.yml file (for Python) that simplifies the installation of these dependencies.

Usability for the Community: The structured format and clarity of the code make it a valuable resource for the community. Additionally, the authors have included scripts for visualizing results, which is an added advantage for those keen on visual feedback.

Reviewer #4 (Remarks to the Author):

Summary of manuscript:

Here the authors present a model with novel architecture --combining existing language model architectures with pretrained protein embeddings -- and training procedure -- with the LDAM loss. This model achieves SOTA MCC on signal peptide prediction, and two versions are made available for public use: one with MSAs and another that is MSA-free. Both versions will be useful for the community, and the authors also provide 347 novel signal peptides for immediate exploration.

Review:

Overall, I found the manuscript to be well-written and the work to be befitting publication in Nature Computational Science already.

One major request is for a few ablations on the novel contributions of this manuscript. Specifically, I am curious about the performance of (Ablation 1) training with a standard cross-entropy objective; and (Ablation 2) swapping the MSA embeddings and ESM embeddings with random inputs.

Otherwise, some minor suggestions are:

* Figure 1b is missing the input to the biLSTM block

* Data in the radar plots of 2d would be appreciated as tables also. The data in Figure 2e would also be appreciated in a supplemental table (apologies if I've missed them).

* Appendix line 255: Include which C hyperparameters you've tried and which one was best.

* Making the datasets available at osf.io instead of as google drive links.

Reviewer #4 (Remarks on code availability):

I have reviewed but `_not_` installed and run the code.

However, I did a quick skim and everything seems to be in order for reproduction.

Author Rebuttal to Initial comments

We are very grateful to the reviewers for their thoughtful and detailed comments, which helped us significantly improve our paper. We have carefully revised the manuscript following all the comments. Below is the point-by-point response to all the reviewers' comments.

 Reviewer #1 (Remarks to the Author):

The paper describes a new tool for the prediction of signal peptides.

Basically, the main novelties reside in

- 1 - the ability to perform prediction without indicating the source organism, that make the method useful for metagenomic studies. In this respect, also SignalP6 is organism-agnostic (as correctly commented in the paper). USPSignal is better performing, although the increment is not so astonishing (as expected, since prediction performance of the state-of-the-art is already very high)
- 2 - the adoption of a balancing procedure to improve the predictive performance. LDAM loss function has been adapted from a previous, still arXived paper, and seems effective.

Answer: Thank you very much for the excellent summary and positive comments! As you have mentioned, USPNNet makes some improvement on the signal peptides prediction performance, and it is able to be applied in metagenomic studies for SP detection. In this revision, we have further improved our manuscript based on your comments.

The writing of the paper and the presentation of the material is often unclear and unfocused and this prevents the complete understanding.

Result presentation is somehow confusing, different plots are presented in figs 2-4 and possibly they should be better organized, improving readability and better defining the flux of information.

Answer: Thank you for this comment. In the 'Result' section, we have modified the beginning of some subsections to make the continuation relationship clearer. Actually, subsections in Result Section have a progressive relationship.

Specifically, for section 2.1, we want to introduce the overview of USPNNet and give some information about the model architecture.

For section 2.2, we present the overall performance of USPNNet and other signal peptide predictors on the benchmark dataset. We analyse the performance of USPNNet for signal peptides of different sequence similarity with training data and different lengths, and the running time of our proposed methods. In addition, we conduct a head-to-head comparison with the SOTA method: SignalP6.0. The introduction of the dataset is not plain, so we rewrite the beginning:

Original version: USPNNet is able to predict the type and cleavage site of a signal peptide at the same time. To fair analyze performance, we train and test our model on the re-classified, extended, and homology-reduced datasets derived from the data published with SignalP5.0 and SignalP6.0. The combination of training and benchmark data is identical to the homology partitioned SignalP6.0 dataset.

Current version: We use the training set and benchmark set published in SignalP6.0 to train and test our model. SignalP6.0 re-classified, extended, and applied homology partitioning to

the SP training dataset published in SignalP5.0 and obtained the new training set. For the benchmark set, SignalP6.0 reused the benchmark set of SignalP5.0, from which they excluded all sequences that were removed in the homology partitioning procedure of the new training set.

For section 2.3, we conduct the ablation study of USPNet, which tests the effectiveness of protein language models, effectiveness of our loss function, and USPNet's group information-independent capacity. To express our intentions more clearly, we have modified the beginning sentences:

Original version: In this part, we will discuss the performances of models with different loss functions and embeddings to look deeply into the reasons why USPNet is universal.

Current version: In this part, we will conduct the ablation study and discuss the performances of our models with various loss functions and protein language model embeddings to look deeply into the effectiveness of different modules in USPNet.

Experiments of section 2.2 and 2.3 are all within the benchmark set, therefore, for section 2.4 and 2.5, we test the generalization of USPNet on some domain-shift data: one independent test set and one proteome-wide study.

And Figure 2, 3, and 4 are results for Section 2.2, 2.3, and 2.4, respectively. In the original manuscript, the order of some of the subfigures in Figure 2 is not the same as the order in which corresponding experiments appear in the text. Therefore, we adjusted the order of writing in Section 2.2. And now, all the Figures are in line with the order of experiments shown in the manuscript.

And in the 'Method' section, the training data and the benchmark data are not well declared in the previous manuscript. We just said they are obtained from SignalP6.0, but not explained how they are generated, and how we applied them in our study. Such explanation may lead to some confusion, therefore, we have rewritten the data description in section 4.1.

Current version: The training and benchmark test set is the same as introduced in SignalP6.0, which reclassified some SP types for data published with SignalP5.0. Further, it added some new SPs from UniProt20 and Prosite21, and new soluble and transmembrane proteins from UniProt and TOPDB22. Then, part of the data in the new dataset was removed with the homology partitioning methodology introduced by Gislason et al.. The homology partitioning procedure partitioned the new dataset containing all protein sequences into three partitions, with sequences in each partition sharing at most 30% sequence identity with sequences from other partitions, and redundant sequences were removed. The resulting dataset is then divided into the training set and the benchmark set. The benchmark set encompasses protein sequences from the original benchmark set of SignalP 5.0 except sequences removed from the homology partitioning procedure of the new dataset. It is worth noticing that since the benchmark set sources some sequences from the same partitions as the training set, there is a small subset within it that exhibits high identity to the training sequences.

authors claim to compare 16 methods, but few of them are actually represented in fig 2 (panels a,b,d, and e - legends include more methods than reported). I agree that MCC and MCC2 are important indexes, but a full evaluation adopting a complete set of scores should be reported in supplementary materials. Also, color scale are difficult to be read and numbers should be provided.

Answer: Thank you for this comment. We are sorry that the number '16' is a typo in the original manuscript, we have tried our best to compare with as many methods as possible and changed it into the correct number '11'. For Sec/SPI type, 9 methods are able to perform prediction. For Sec/SPII, 5 methods can do it, for Tat/SPI, 6 methods can do it, and for Tat/SPII, only 4 methods can do it ([Figure 2.d](#)).

In addition, besides MCC1 and MCC2, we also added the other evaluation metrics, including Balanced Accuracy, Precision, Recall, and F1-score for each SP type. The results of the added evaluation metrics could be found in Supplementary Table 5-9. We list one table here for your reference:

Supplementary Table 5: Benchmarking of Sec/SPI signal peptide detection predictions

Method	Archaea				Eukaryotes			
	Precision	Recall	F1 score	BA(Balanced Accuracy)	Precision	Recall	F1 score	BA(Balanced Accuracy)
SignalP6.0	0.947	0.692	0.8	0.841	0.702	0.922	0.797	0.956
DEEPSIG	n.d.	n.d.	n.d.	n.d.	0.769	0.783	0.776	0.888
LipoP	0.63	0.654	0.642	0.771	0.234	0.513	0.322	0.735
PRED-LIPO	0.581	0.692	0.632	0.773	0.161	0.226	0.188	0.598
PRED-SIGNAL	0.565	1.0	0.722	0.888	0.147	0.487	0.226	0.708
PRED-TAT	0.647	0.846	0.733	0.856	0.299	0.525	0.319	0.698
TOPCONS2	0.444	0.615	0.516	0.695	0.294	0.878	0.44	0.913
USPNet-fast	0.944	0.654	0.773	0.821	0.829	0.887	0.857	0.941
USPNet	1.0	0.654	0.791	0.827	0.872	0.887	0.879	0.942

And to make the figures clearer and informative, we replaced the radar plots in [Figure 2.d](#) into bar plots, and changed the color scale of the heatmap in [Figure 2.e](#). And the detailed numbers of results could be found in Supplementary Table 1-4.

Benchmark dataset seems to include proteins sharing a large sequence identity with the training set (Fig 2b, up to 90%).

Fair evaluation and comparison requires either to reduce the level of similarity or to rely on the independent testing set (whose evaluation and comparative assessment reported in fig 4 is quite limited)

Answer: Thank you for this comment that help us to make our experiment more solid. In the original version of our manuscript, we actually conducted experiments on benchmark set that reduced the sequence similarity. We totally tested 7 different cut-off values (30% to 90%), for each cut-off value, take 30% as an example, we remove all the sequences in the benchmark set which have more than 30% similarity to the sequences in the training set. The result is shown in Figure 2.b. However, the caption and legend of Figure 2.b is unclear and may leads to some misunderstanding. The readers may misconsider that we only group the sequences by similarity, but no similarity cut-off test is done. Therefore, we have modified the caption and legend of [Figure 2.b](#), the new one is shown below:

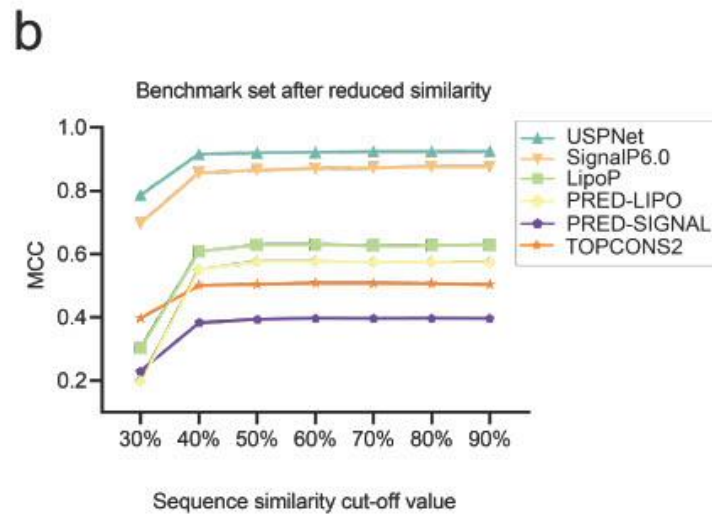


Figure 2.b: The performance of the benchmark set at different similarity cut-off values. We tested 7 different cut-off values and USPNet consistently outperforms other models on MCC.

Besides, we acknowledge that the comparison requires to reduce the level of similarity, hence, in the revised manuscript, all the results in Figure 2.d and Figure 2.e are based on the new benchmark set with a cut-off value of 40%. Specifically, we remove all the sequences in the benchmark set which have more than 40% similarity to the sequences in the training set, and the amount of data in the new benchmark set dropped from 6611 to 5292. The new Figure 2.d is shown below:

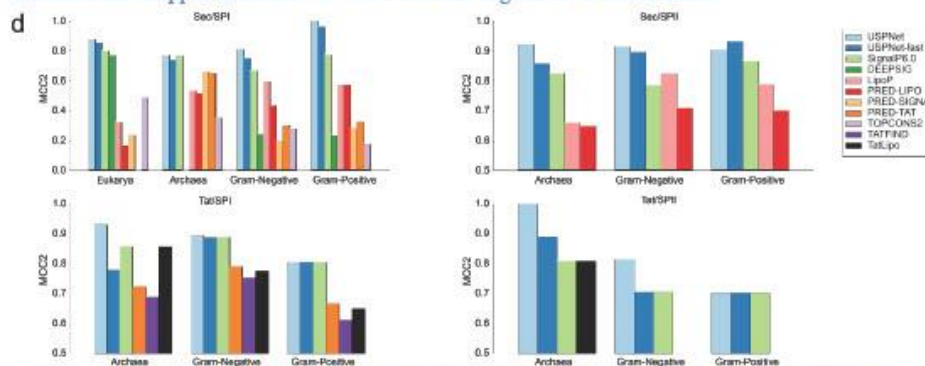


Figure 2.d: Bar plots of USPNet and other models on different perspectives of performance. We compare MCC2 of different models from the viewpoint of organism groups and SP type. USPNet behaves as the most powerful signal peptide predictor in every organism group.

USPNet demonstrates superior performance on both MCC2 and MCC1, especially for data with Tat/SPI label, our method outperforms others for at least 10% across all 3 organism groups. And for the type with the most data (Sec/SPI), USPNet keeps its performance as the best one among all the included models (detailed number could be found at Supplementary table 1-4). And even our USPNet-fast does better than all other competitors. Our conclusion in the original manuscript still holds: USPNet has good generalization and outperforms other models on SP prediction task.

The evaluation of the fast method in relation to the MSA based one is reported only for discrimination (fig 2d) and not for cleavage site prediction. Still, fig 2d is unclear and it is difficult to assess the effective performance of the method.

Answer: Thank you for the comment. In the original manuscript, we included the cleavage site results of USPNet, SignalP5.0 and SignalP6.0, and did not show the results of USPNet-fast. In the revised manuscript, we have changed the cleavage site prediction result in Figure 2.e, to make it consistent with the methods we included in the SP type classification result in Figure 2.e (USPNet, USPNet-fast, and SignalP6.0), so that we could directly see the detailed comparison between USPNet and USPNet-fast in the cleavage site prediction. The new [Figure 2.e](#) is shown below, and the detailed numbers of results for cleavage site prediction could be found at Supplementary table 15-18:

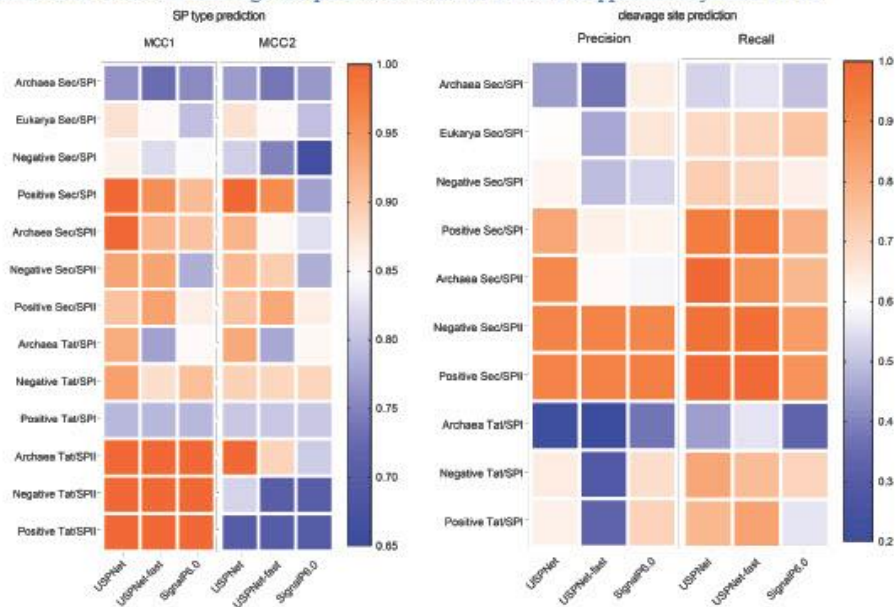


Figure 2.e: Comparison between USPNet, USPNet-fast, and SignalP6.0 on both signal peptide type prediction and signal peptide cleavage site prediction.

From the results, we could see that the cleavage site prediction performance of USPNet-fast is comparable with that of USPNet.

And to make [Figure 2.d](#) clear and more informative, we have changed the radar plots into bar plots. The detailed numbers of the results of [Figure 2.d](#) and [Figure 2.e](#) could be found at Supplementary Table 1-4.

Authors introduce the possibility for the user to "adjust weights according to their specific needs, enhancing USPNet's effectiveness and suitability for their application scenarios." This is mostly unclear: on which basis a user can change parameters? and how, operatively?

Answer: Thank you for pointing out it. We realize that our expression here can lead to the readers feel confusing, because they do not know exactly how to operate that. Therefore, for the convenience of users, we are preparing to provide more trained models that could focus on the prediction of the major classes (Sec/SPI) to satisfy different users' demand. The instructions of how to use these models could be found in our [github repo, README file](#).

In addition, to make the manuscript easy to read, we deleted this sentence. Instead, we put the content that related to tuning process of class weight hyperparameters in Section 4.2, Training details, for the advanced users to understand the hyperparameters we used for training, the added content is as shown below:

Current version: We use LDAM loss as the objective function in training. For the signal peptide prediction objective function, L_s , class-balanced re-weighting is utilized to assign weights based on the inverse of sample frequencies for each class. In the cleavage site prediction objective function, L_c , a heightened weight of 6 is attributed to the class signifying the cleavage site, the class representing paddings is assigned with weight of 0, while a standard weight of 1 is assigned to all other classes.

A few questions are worth an investigation (not essential):

- Models adopted for encoding generally depends on the whole sequence, bidirectionally. What is the effect of truncating the sequence at 70 residues (pre-encoding) instead of truncating the encoding resulting from the whole sequence?

Answer: Thank you for asking the very insightful question! We developed the model for signal peptide prediction, and SPs are short and located in the N-terminus of proteins. Therefore, we want to make the model focuses on the front part of the protein sequence. In Supplementary Figure 1, we have counted the length of the proteins, as well as their corresponding signal peptides, in our dataset, and found that most SPs (more than 99%) are less than 50 in length. However, the proteins vary in length, with some exceeding 1000. If we want to input the whole sequence into the Bi-LSTM model, we need to unify the length of these inputs (padding) to make sure they have the same shape. Such action is not helpful if we want to focus on the N-terminus of proteins. Hence, we choose the input length as 70, which can cover all signal peptides. We found that this operation would introduces another problem: we cannot encode the whole protein sequences, which will lead to the absence of sequence information. To resolve the problem, we utilize the embeddings generated by the protein language model, these embeddings encode not only the complete sequence information of proteins, but also the evolutionary information and structural information of proteins. The application of protein language model embeddings significantly improves the generalization of our model ([Figure 3.b and .c](#)).

b

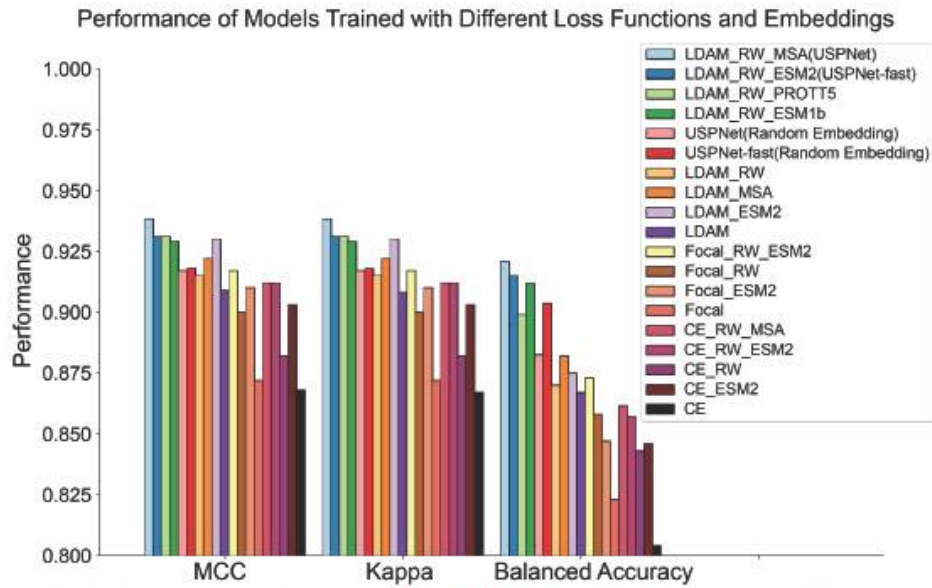


Figure 3.b: Ablation study performance of USPNet: MCC, Kappa, and Balanced Accuracy of models trained with different loss functions and protein language model (PLM) embeddings (LDAM, Focal, and CE mean 3 different loss functions. RW means using class reweighting. MSA and ESM2 are the PLM embeddings of USPNet and USPNet-fast; PROTT5 and ESM1b are embeddings obtained from the other 2 PLMs for comparison. Random Embeddings means replacing PLM embeddings with random inputs).

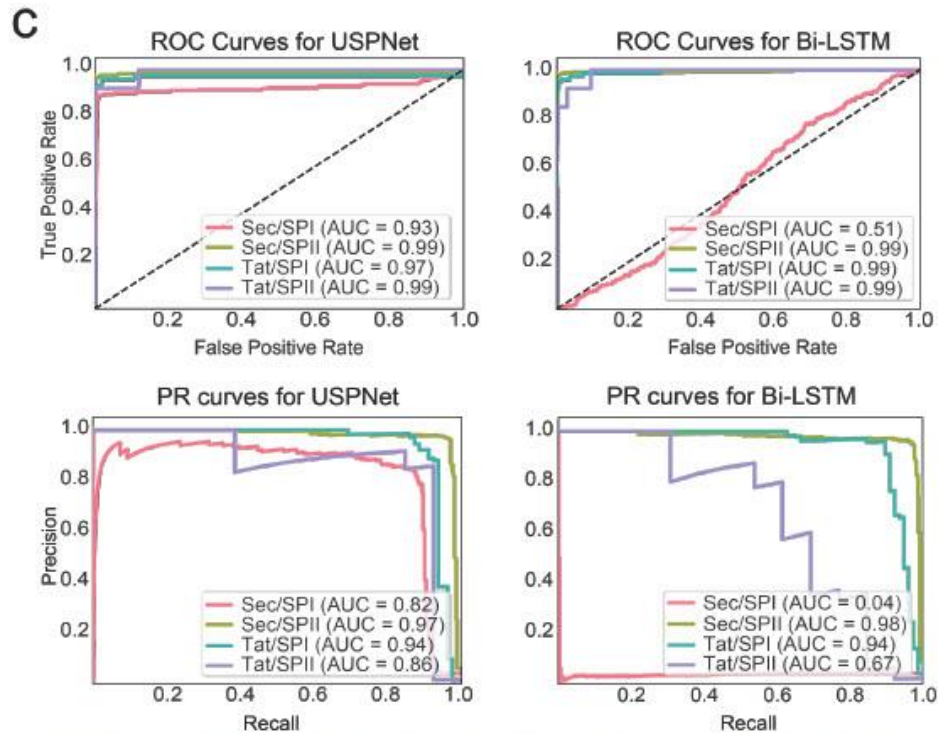


Figure 3.c: ROC curves and Precision-Recall curves of USPNet and Bi-LSTM on different types of signal peptides prediction. Our model is stable across all 4 kinds of SPs, while Bi-LSTM suffers from the data imbalance problem.

- In metagenomic studies sometimes assembled contigs show truncations at the N-ter. How robust is the method in this case?

Answer: Thank you for raising the excellent question! In our novel signal peptide discovery pipeline, the last step is metaproteome filtering, where we only retain the proteins that can actually express. To further investigate the influence of truncations at the N-terminus, we carry out the experiment that truncates different lengths of some sequences' N-terminus to see the performance of the SP prediction. According to previous research (Gibson, Bond et al. 2017), the change of amino acid at position -1, -2, and -3 of signal peptides can impact the cleavage site. Therefore, to avoid such impact, we only truncate the first few amino acids. Specifically, we select all 735 proteins that have a signal peptide from our benchmark set and truncate the first 1, 2, 3, 4, and 5 amino acids of their SPs, respectively. We counted the number of correctly detected signal peptides, the result is shown in the table:

	Correctly predicted number/Total number	Accuracy
Original SP	673/735	0.916
truncate 1AA	663/735	0.902
truncate 2AAs	664/735	0.903
truncate 3AAs	656/735	0.893
truncate 4AAs	648/735	0.882
truncate 5AAs	637/735	0.867

From the results, we could see that the performance decreased, but the degradation is not huge. Even if we truncate 5 AAs of the signal peptide, USPNet is still able to correctly predict most of the SPs. The experiment demonstrates that our model could handle the case when truncation occurs, and it further proves the robustness of USPNet.

Minor:

spell check is needed: just two examples from the first sentences in the introduction

Secretary --> secretary

Von Heijen --> Von Heijne

Hidden Markov models can be also used as supervised models

Answer: Thanks for this comment. We have thoroughly checked and corrected the grammatical errors and typos we found in our revised manuscript. And to avoid ambiguity, we remove the sentence in line 47 with 'Hidden Markov model':

Original version: Furthermore, generative models, such as the hidden Markov model (HMM), were proposed to facilitate the recognition of signal peptides.

Current version: Furthermore, generative models were proposed to facilitate the recognition of signal peptides.

Reviewer #2 (Remarks to the Author):

MAJOR REVISIONS:

After reading the whole paper, it is not clear to me how the datasets were used to train and evaluate the method.

In the Methods section, three datasets are detailed, namely: i) training, ii) benchmark test and iii) independent test.

Answer: Thank you for mentioning these 3 datasets we used in our study. Here is the brief introduction about them.

- Training set: The same as introduced in SignalP6.0, we used 5-fold cross validation on training set to select the hyperparameters.
- Benchmark set: The original benchmark set is the same as introduced in SignalP6.0, to better evaluate our model, we curated a new benchmark set with a similarity cut-off value of 40% to the training set. The results in [Figure 2.d](#) are tested on the new benchmark set.
- Independent test: We want to further evaluate the generalization of USPNet on a more rigorous dataset, therefore, we collected some data with following steps: (1) we remove proteins before November 2020 (the date of SignalP6.0 dataset collection), and proteins composed of less than 30 amino acids; (2) we select those from eukaryotes, Gram-positive and Gram-negative bacteria; (3) we select proteins containing signal peptides with a confident label ECO:0000269 (experimental annotation) and ECO:0000305 (manually curated annotation) from the Swiss-Prot database (released on 2022/04). Furthermore, to better ensure the independence of the SP22 dataset, CD-HIT is applied to remove redundant proteins sharing more than 40% similarity with proteins in our training dataset. And these data make up the independent test set, which is more challenging than the benchmark set.

And the detailed description of our training set and benchmark set could be found in our following answers. We also revised the corresponding sections in the manuscript to make it more clear.

In line 186, the authors report that "Besides the benchmark set, we also conduct 5-fold cross-validation on the full training and benchmark sets (Supplementary Table 5-9). Since homology partitioning is inoperative in cross-validation, USPNet showcased more impressive performance, especially for the minor classes such as Tat/SPII and Sec/SPIII signal peptides."

Answer: Thanks for this comment and sorry for the unclear statement in the text. In the 'Method' section, the training data and the benchmark data are not well declared in the previous version of the manuscript and lead to some confusion, therefore, we have rewritten the data description in section 4.1.

Current version: The training and benchmark test set is the same as introduced in SignalP6.0, which reclassified some SP types for data published with SignalP5.0. Further, it added some new SPs from UniProt20 and Prosite21, and new soluble and transmembrane proteins from UniProt and TOPDB22. Then, part of the data in the new dataset was removed with the homology partitioning methodology introduced by Gislason et al. The homology partitioning procedure partitioned the new dataset containing all protein sequences into three partitions, with sequences in each partition sharing at most 30% sequence identity with sequences from other partitions, and redundant sequences were removed. The resulting dataset is then divided into the training set and the benchmark set. The benchmark set encompasses protein sequences from the original benchmark set of SignalP 5.0 except sequences removed from the homology partitioning procedure of the new dataset. It is worth noticing that since the benchmark set sources some sequences from the same partitions as the training set, there is a small subset within it that exhibits high identity to the training sequences.

Besides, our intention for conduct 5-fold cross-validation on the combined training and benchmark sets is that the benchmark set has no data of type Sec/SPIII, thus we also want to evaluate the performance of USPNet on this type. We understand that the description about this cross-validation is not clear, therefore, we rewrite this part:

Current version: As our benchmark set has no signal peptides in the type of Sec/SPIII. Therefore, to comprehensively assess the robustness and consistency of our model on Sec/SPIII signal peptides, we also employed 5-fold cross-validation on the combined training and benchmark sets (Supplementary Table 10-14). Since homology partitioning is inoperative in cross-validation, USPNet showcased more impressive performance, especially for the minor classes such as Tat/SPII and Sec/SPIII signal peptides.

Moreover, from Figure 2B, it seems like the benchmark dataset is not homology reduced against the training dataset as there are sequences with up to 90% similarity.

Answer: Thank you for pointing out the problem of our presentation that help us to make our experiment more clear and solid. In the original version of our manuscript, we actually conducted experiments on benchmark set that reduced the sequence similarity. We totally tested 7 different cut-off values (30% to 90%), for each cut-off value, take 30% as an example, we remove all the sequences in the benchmark set which have more than 30% similarity to the sequences in the training set. The result is shown in Figure 2.b. However, the caption and legend of Figure 2.b is unclear and may leads to some misunderstanding. The readers may misconider that we only group the sequences by similarity, but no similarity cut-off test is done. Therefore, we have modified the caption and legend of Figure 2.b, the new one is shown below:

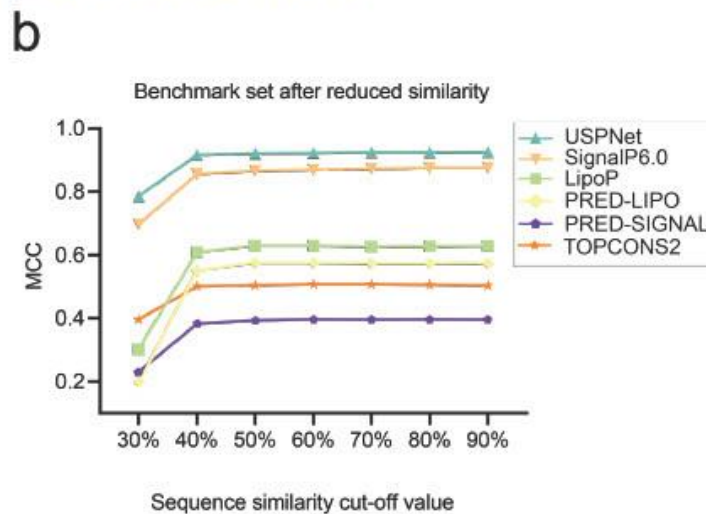


Figure 2.b: The performance of the benchmark set at different similarity cut-off values. We tested 7 different cut-off values and USPNet consistently outperforms other models on MCC.

Besides, we acknowledge that the comparison requires to reduce the level of similarity, hence, in the revised manuscript, all the results in Figure 2.d and Figure 2.e are based on the new benchmark set with a cut-off value of 40%. Specifically, we remove all the sequences in the benchmark set which have more than 40% similarity to the sequences in the training set, and the amount of data in the new benchmark set dropped from 6611 to 5292. The new Figure 2.d is shown below:

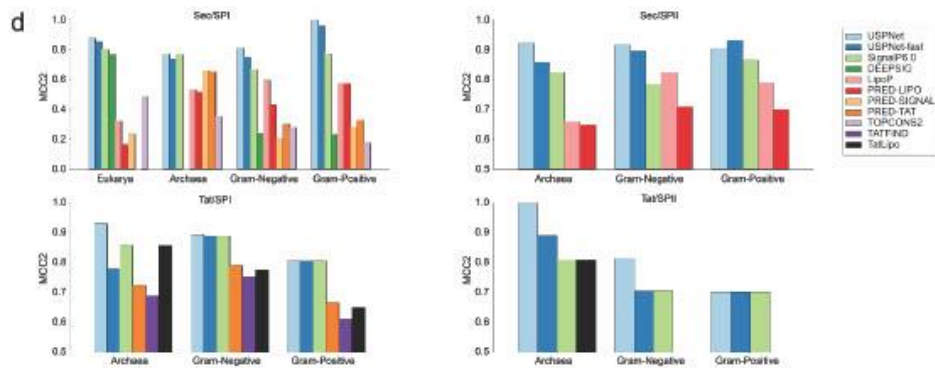


Figure 2.d: Bar plots of USPNet and other models on different perspectives of performance. We compare MCC2 of different models from the viewpoint of organism groups and SP type. USPNet behaves as the most powerful signal peptide predictor in every organism group.

USPNet demonstrates superior performance on both MCC2 and MCC1, especially for data with Tat/SPI label, our method outperforms others for at least 10% across all 3 organism groups. And for the type with the most data (Sec/SPI), USPNet keeps its performance as the best one among all the included models (Supplementary table 1-4). And even our USPNet-fast does better than all other competitors. Our conclusion in the original manuscript still holds: USPNet has good generalization and outperforms other models on SP prediction task.

My understanding is that the benchmark dataset was adopted to select the model architecture and hyperparameters in a cross-validation setting, as well as for reporting results in Figure 2 to benchmark against other methods. The independent test dataset is instead only adopted to report results in section 2.5.

If this is the case, it would be better to report only comparative results against other methods on the completely independent blind test set and leave the so-called benchmark dataset as a validation dataset adopted to select the model. If my understanding is not correct, it would be better to rephrase the paper to better explain the training and validation procedures

Regarding USPNet-fast, the authors should try ProtT5 and ESM2 instead of ESM-1b, which generally performs worse, and they could provide comparisons between the different language models.

Answer: Thanks for this comment. Actually, the benchmark set we used is the same as that of SignalP6.0, the benchmark set applied the homology partitioning based on the benchmark set provided by SignalP5.0, but there are still few sequences that have high similarity against sequences in the training set. Therefore, to better compare the performance of different methods, we now use the new benchmark set with a cut-off value of 40% (the details could be found in the last answer). To make fair comparison, we train USPNet, USPNet-fast, and SignalP6.0 on the training set, and test these methods on the benchmark set. For the hyperparameters selection, we actually applied the 5-fold cross validation on the training set. Specifically, we randomly divided the training set into 5 folds, and each time leave one fold for validation and the other 4 folds for training to tune hyperparameters. After the hyperparameters were determined, we then used the whole training set to train USPNet.

Therefore, the benchmark set can be used to test the performance of these methods. We are sorry that our model hyperparameters selection process is not clearly described, so we added hyperparameters selection details in Section 4.2 training details. And the hyperparameters we used to train the model could also be found in this section.

Section 4.2: For hyperparameters selection, we use 5-fold cross-validation on the training set. Specifically, we divide the training data into 5 subsets, and then systematically train and evaluate the model five times. In each iteration, one of the folds is used as the validation set while the remaining four folds are used for training. This process is repeated five times, with each fold taking a turn as the validation set. We tune the hyperparameters based on the average performance obtained from these iterations. Then, at the training stage, the complete training set is used for training.

Thank you very much for the excellent idea that helps us further improve the performance of USPNet-fast. For the USPNet-fast, we have tried ProtT5 and ESM2, and found that ESM-2 has the best performance among all three single sequence protein language models. Therefore, in the revised manuscript, we have changed the PLM module of USPNet-fast from ESM-1b to ESM-2, and the results of USPNet-fast on the benchmark set is modified accordingly (Figure 2.c, d, e). In addition, we also renewed the results of USPNet-fast in ablation study (Figure 3.b). We make thorough comparison of different settings for the protein language module; the result is shown below:

b

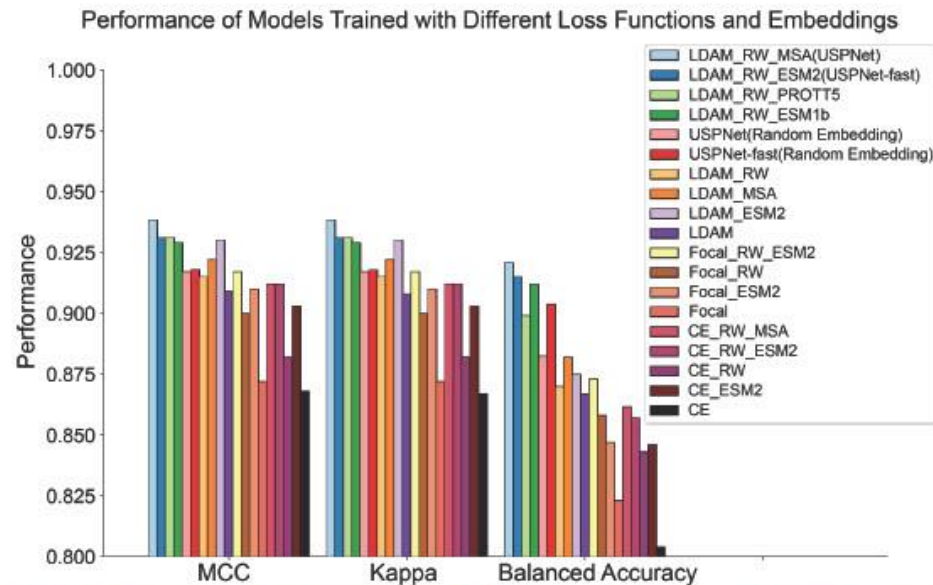


Figure 3.b: Ablation study performance of USPNet: MCC, Kappa, and Balanced Accuracy of models trained with different loss functions and protein language model (PLM) embeddings (LDAM, Focal, and CE mean 3 different loss functions. RW means using class reweighting. MSA and ESM2 are the PLM embeddings of USPNet and USPNet-fast; PROTT5 and ESM1b are embeddings obtained from the other 2 PLMs for comparison. Random Embeddings means replacing PLM embeddings with random inputs).

	ESM2	ProtT5
MCC(Overall)	0.931	0.931
Kappa	0.931	0.931
Balanced Accuracy	0.915	0.899

In the [figure](#), LDAM_RW_ESM2 is USPNet-fast, LDAM_RW_PROTT5 is the model replace ESM-2 of USPNet-fast into ProtT5, and LDAM_RW_ESM1b is the model replace ESM-2 of USPNet-fast into ESM-1b We could see that for MCC and Kappa, three language models have comparable performance, however, when look into the Balanced Accuracy, ProtT5 is far behind other two models, meaning that ProtT5 is not efficient enough in predicting signal peptides belong to minor classes. In addition, we also test the model without PLM embeddings (like LDAM_RW, LDAM in the figure), and the model that replace the PLM embeddings into random inputs (USPNet(Random Embedding) and USPNet-fast(Random Embedding)) to further evaluate the effectiveness of PLM embeddings.

MINOR REVISIONS:

Figure 1B is confusing as the arrows go in different directions and it is not easy to follow the pipeline of the method

Figure 2D, radar plots are not Very informative

Answer: Thanks for this comment. The original version of Figure 1B has different styles of arrows, making it confusing, hence, we have corrected the arrows in Figure 1B to make them in the same style. We have also added some arrows to make the data flow clearer. The revised version of Figure 1B is shown below:

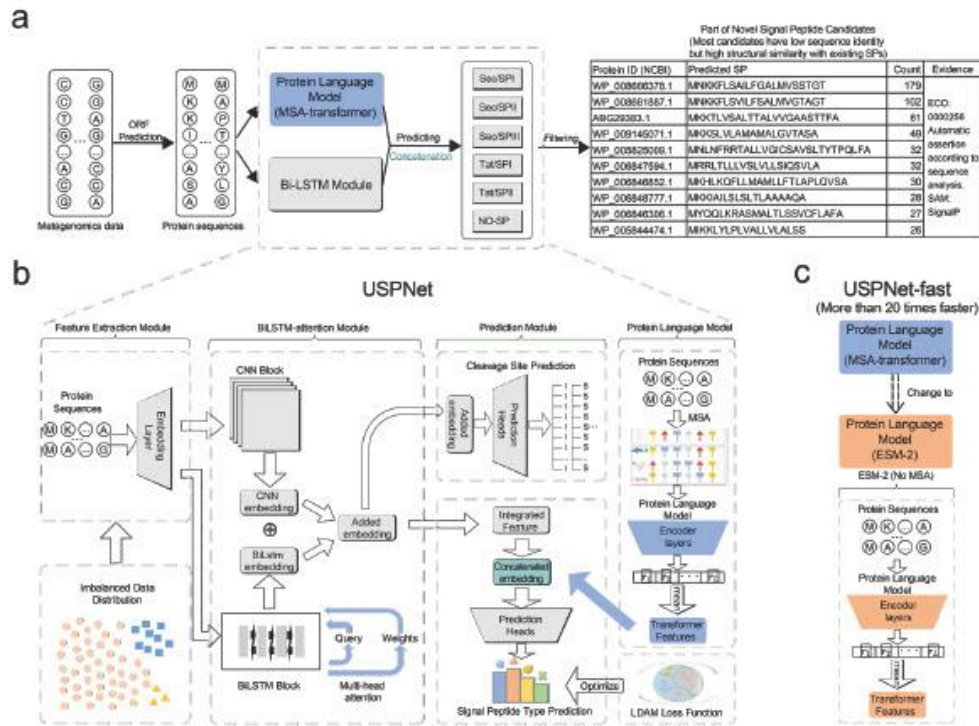


Figure 1.b: Detailed architecture of USPNet. The training data is imbalanced. The protein sequences go through the feature extraction module and are then passed to the BiLSTM module, which includes a Bi-LSTM layer with self-attention and a CNN for extracting long-distance dependencies and features of the sequences. For SP type prediction, USPNet incorporates MSA embeddings generated by a pre-trained MSA Transformer model. Subsequent MLP-based modules predict cleavage sites and signal peptide types. Label Distribution-Aware Margin (LDAM) loss is employed in training to address data imbalance.

For Figure 2D, because we include a lot of methods, the radar plots are not easy to see. In the revised version, we replace the radar plots into bar plots, as shown below:

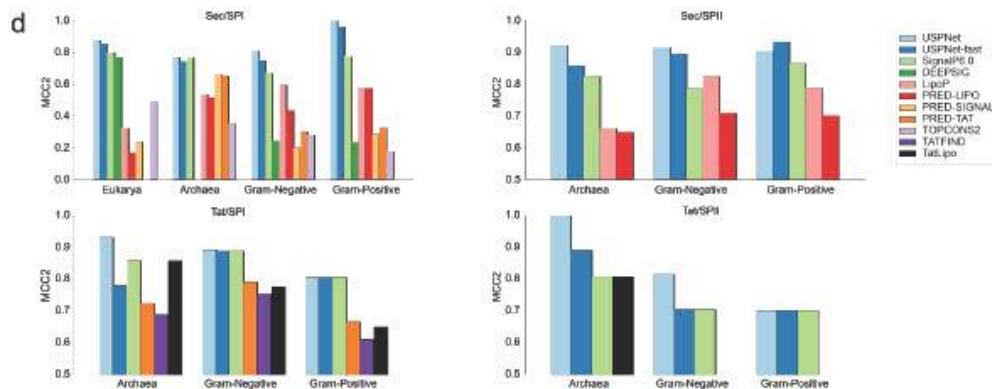


Figure 2.d: Bar plots of USPNet and other models on different perspectives of performance. We compare MCC2 of different models from the viewpoint of organism groups and SP type. USPNet behaves as the most powerful signal peptide predictor in every organism group.

For Sec/SPI type, 9 methods are able to perform prediction. For Sec/SPII, 5 methods can do it, for Tat/SPI, 6 methods can do it, and for Tat/SPII, only 4 methods can do it.

Reviewer #2 (Remarks on code availability):

I went to what is in the link and this is correct. I did not install the code for lack of time and for difficulties with my laptop.

Reviewer #3 (Remarks to the Author):

The authors have developed a Bi-LSTM-based method, USPNet, for predicting the types of signal peptides. USPNet incorporates protein language model and specially designed loss function to resolve the group information dependency and data imbalance problem in signal peptide prediction. Authors then build a pipeline and try to use genome data to discover novel signal peptides. They finally provide 347 sequences that are likely to be novel SPs. Some of the SPs have been supported by literature.

The comprehensive experiments prove USPNet has good generalization, and USPNet has good performance even if the sequence similarity is low. It is effective in domain-shift data as well as proteome-wide study. The methodology is decent to be applied outside of the SP field since data imbalance problem is common in protein-related tasks. The main contribution of this work is the authors develop a pipeline to discover potentially novel signal peptides from porcine gut, it started from the genome data and ends up with the signal peptide candidates. The pipeline is handy and can produce plenty of candidate sequences at a reasonable time. It is quite interesting that although the model just takes amino acid sequences as input and does not directly include protein tertiary structure

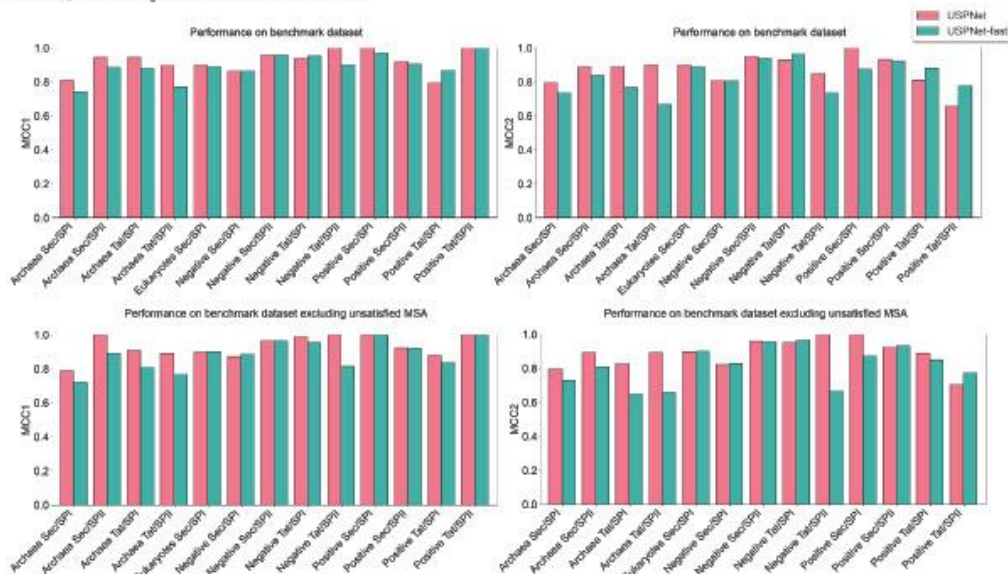
information during the training stage, it screens the candidate sequences mainly based on the structural similarity instead of the sequence similarity. And the literature-based evaluation provides evidence for their experimental conclusions.

The manuscript is written in a clear and understandable manner, and the results are consistent with authors' conclusions. I have a few suggestions to improve the quality of the manuscript.

Answer: We truly appreciated your thorough and constructive review of the paper and for taking the time to go through many of the details! As you have mentioned, it is pioneering that we try to discover novel SPs from swine gut data, and finally provide 347 candidates. Based on your comments, our manuscript has been revised and improved. Below we respond to all your concerns point by point.

1) In Figure 2.e the authors mentioned that the incorrect predictions normally have poor Neff scores. If possible, authors might compare USPNet with USPNet-fast by excluding these proteins with unsatisfied MSA quality.

Answer: Thanks for this comment that helps us to analyse the performance improvement when excluding the unsatisfied MSA. According to Rao et al.'s research (Rao, Liu et al. 2021), MSA transformer outperforms ESM-1b when using 16 input sequences. Therefore, we add an experiment that excluded all the datapoints with Neff score less than 15 and finally get 4996 sequences (originally 6661 sequences) to be our new dataset. We test the MCC of USPNet and USPNet-fast on the new dataset, the comparison is shown below:



The overall MCC of USPNet increases from 0.938 to 0.943. In some types, the improvements are obvious, especially for the Gram+ Tat/SPI, USPNet is behind USPNet-fast for about 7% on both MCC1 and MCC2 in the original benchmark set, however, in the new dataset that excluded the

unsatisfied MSA, USPNet increased from to 0.877 on MCC1 and from 0.806 to 0.887 on MCC2, which outperform USPNet-fast by 4.2% and 4.0%, respectively. In addition, USPNet has superior performance for almost all the types of SP in the new dataset. There is clear evidence that high-quality MSA enables better SP identification performance of USPNet. We have added the experiment and result in the supplementary file Section 3.

2) There is a lack of Precision and Recall of USPNet-fast in Figure 2.e. The result should be provided to show the effectiveness of USPNet-fast in cleavage site prediction.

Answer: Thanks for this comment. In the revised manuscript, we have changed the cleavage site prediction result in Figure 2.e, to make it consistent with the methods we included in the SP type classification result in Figure 2.e (USPNet, USPNet-fast, and SignalP6.0), so that we could directly see the detailed comparison between USPNet and USPNet-fast in the cleavage site prediction. The new Figure 2.e is shown below:

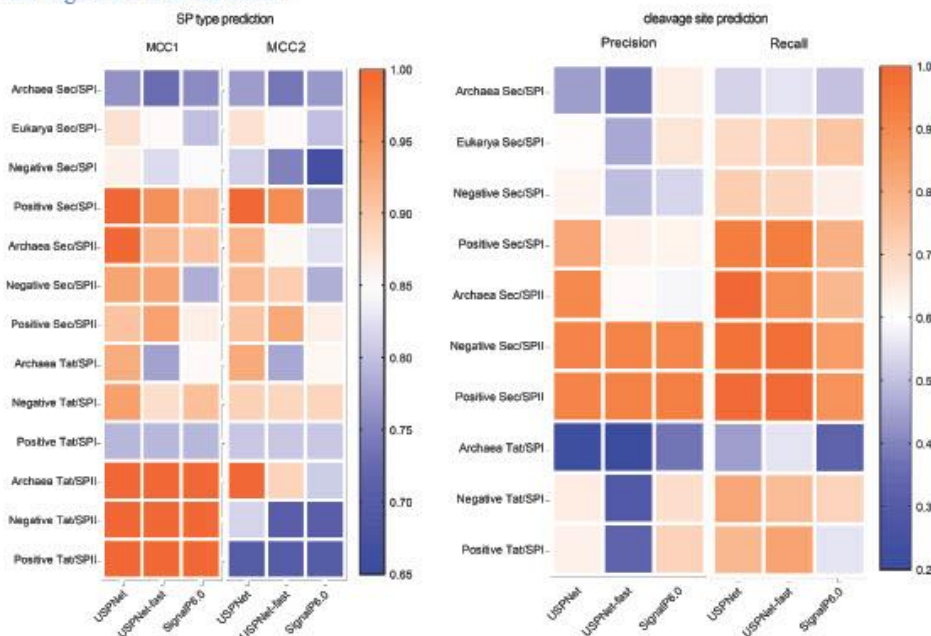


Figure 2.e: Comparison between USPNet, USPNet-fast, and SignalP6.0 on both signal peptide type prediction and signal peptide cleavage site prediction.

From the results, we could see that the cleavage site prediction performance of USPNet-fast is comparable with that of USPNet.

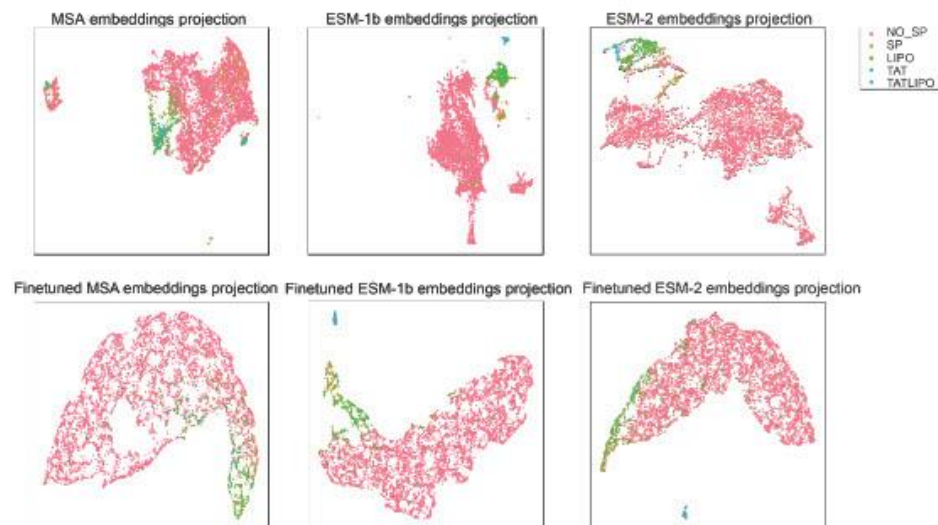
3) Line 94, 'discovered' -> 'discover'.

Answer: Thanks for this comment. We have thoroughly checked and corrected the grammatical errors and typos we found in our revised manuscript.

4) In Figure 3.a, for the MSA and ESM-1b embeddings, are they obtained from the original pre-trained model or from the model fine-tuned on the signal peptide training data? Authors might provide both for comparison to show if there is any change after fine-tuning.

Answer: Thanks for this comment. We are sorry for not explaining it in our manuscript. In Figure 3.a, the embeddings are obtained from the original pre-trained model. The reason we display the protein language model embedding projection is to explore if the original embedding itself includes information related to the types of signal peptide. In addition, we intended to compare the protein language model embeddings with USPNet embeddings to see how well USPNet encodes the SP types information. This question reminded us that we ignore to compare the original embeddings of PLM with embeddings after trained by our SP data. Therefore, we obtained the trained embeddings from the last layer of the protein language model module of USPNet and still apply UMAP for data dimensionality reduction.

Besides the results of MSA transformer and ESM-1b, we also added the embeddings of ESM-2, since we found that for USPNet-fast, ESM2 performs better than ESM-1b on SP prediction. And the result of projection is shown below:



We found that after fine-tuning, the type 'TATLIPO' forms a distinct cluster away from other classes, and for other classes, we can still see demarcation. Due to the loss function we used, the fine-tuning process help to generate embeddings that can better distinguish these minor classes. And to clarify the embeddings we used in Figure 3.a, we modified the sentence in line 236:

original version: For the two protein language models applied in our study to build USPNet and USPNet-fast, we take their output directly for visualization.

current version: For the two protein language models applied in our study to build USPNNet and USPNNet-fast, we take their output from the original pre-trained model directly for visualization.

5) The authors miss some spaces between text and citations/brackets.

Answer: Thanks for this comment. We have thoroughly checked and added the spaces between text and citations/brackets in our revised manuscript.

6) Figure 5.a, a typo: 'Swine gut meragenome collection'

Answer: Thanks for this comment. We have thoroughly checked and corrected the grammatical errors and typos we found in our revised manuscript.

Reviewer #3 (Remarks on code availability):

Reproducibility of Results: users can reproduce the results presented in the paper using the provided code and datasets. The results were largely consistent with those described in the manuscript with minor variations that can be attributed to stochastic elements of the model. The differences were within acceptable margins, implying that the research is indeed replicable.

The code repository includes a comprehensive README file that provides clear instructions on the prerequisites, installation process, and steps to execute the code.

All necessary dependencies are listed. Authors have provided an environment.yml file (for Python) that simplifies the installation of these dependencies.

Usability for the Community: The structured format and clarity of the code make it a valuable resource for the community. Additionally, the authors have included scripts for visualizing results, which is an added advantage for those keen on visual feedback.

Reviewer #4 (Remarks to the Author):

Summary of manuscript:

Here the authors present a model with novel architecture --combining existing language model architectures with pretrained protein embeddings -- and training procedure -- with the LDAM loss. This model achieves SOTA MCC on signal peptide prediction, and two versions are made available for public use: one with MSAs and another that is MSA-free. Both versions will be useful for the community, and the authors also provide 347 novel signal peptides for immediate exploration.

Review:

Overall, I found the manuscript to be well-written and the work to be befitting publication in Nature Computational Science already.

Answer: We are very grateful for your insightful summary and support for our work! As you mentioned, our main contributions are utilizing the protein language models, designing a task-specific loss function, and discovering some novel SP candidates.

One major request is for a few ablations on the novel contributions of this manuscript. Specifically, I am curious about the performance of (Ablation 1) training with a standard cross-entropy objective; and (Ablation 2) swapping the MSA embeddings and ESM embeddings with random inputs.

Answer: Thanks for this comment. In Figure 3.b of our original manuscript, we have tested 4 models using cross-entropy loss. They are USPNet-fast with CE loss (CE_RW_ESM2 in Figure 3.b), USPNet-fast with CE loss and using the default class weight (CE_ESM2), Bi-LSTM with CE loss (CE_RW), and Bi-LSTM with CE loss and using the default class weight (CE). But we ignored to provide the ablation result for USPNet training with CE loss (CE_RW_MSA). So in the revised manuscript, we add the result. The overall MCC, Kappa, and Balanced Accuracy of 'CE_RW_MSA' are 0.912, 0.912, and 0.862, respectively. Compare with the USPNet, whose overall MCC, Kappa, and Balanced Accuracy are 0.938, 0.938, and 0.920, respectively, the cross-entropy loss is not as effective as LDAM loss in handling class imbalance problem.

And for the Ablation 2, in our original manuscript, we have only tested the model without embeddings obtained from protein language model (LDAM_RW). In the revised manuscript, we added an experiment that replace the protein language model embeddings with the random inputs. Specifically, since the MSA-transformer embeddings are 768-dimensional vectors, each with a value between -10 and 10. The random inputs are also set to be 768-dimensional vectors and the value for each dimension is a random number from -10 to 10. We called this model with random embeddings as 'USPNet (Random Embedding)'. The overall MCC, Kappa, and Balanced Accuracy of 'USPNet (Random Embedding)' are 0.917, 0.917, and 0.882, respectively, and its performance is still far behind USPNet. Interestingly, the performance of 'LDAM_RW' (USPNet that remove the protein language model module) on these three evaluation metrics are 0.918, 0.918, and 0.874, which are comparable to that of 'Random Embedding'. For USPNet-fast, the ESM embeddings are 1280-dimensional vectors, we also replace them with 1280-dimensional random vectors, and the model is called USPNet-fast (Random Embedding), and the results are similar with that of USPNet (Random Embedding).

The results indicate that the random embedding input neither disturbs nor improves the performance of the model. We suspect that this is because the linear layers on the top of the PLM module helps fit the signal peptide data and prevent performance degradation.

According to the new results, we have updated [Figure 3.b](#). In addition, we found that ESM-2 could improve the performance of USPNet-fast compared with ESM-1b, therefore, we change the protein language model module of USPNet-fast from ESM-1b to ESM-2, that's why some models in [Figure 3.b](#) are named ESM-2.

b

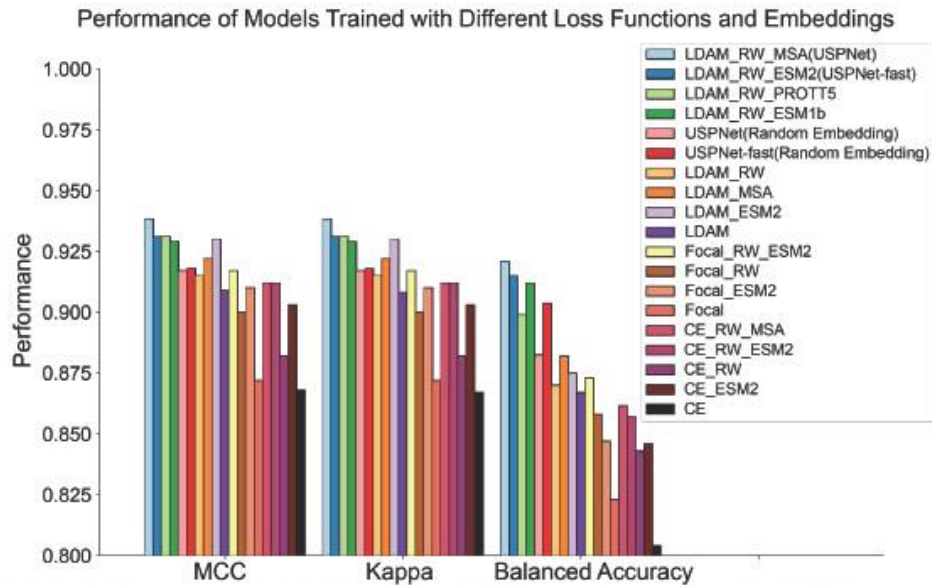


Figure 3.b: Ablation study performance of USPNet: MCC, Kappa, and Balanced Accuracy of models trained with different loss functions and protein language model (PLM) embeddings (LDAM, Focal, and CE mean 3 different loss functions. RW means using class reweighting. MSA and ESM2 are the PLM embeddings of USPNet and USPNet-fast; PROTT5 and ESM1b are embeddings obtained from the other 2 PLMs for comparison. Random Embeddings means replacing PLM embeddings with random inputs).

Otherwise, some minor suggestions are:

* Figure 1b is missing the input to the biLSTM block

Answer: Thanks for this comment. The original version of Figure 1B has different styles of arrows, making it confusing, hence we have corrected the arrows in Figure 1B to make them in the same style. In addition, we also add the arrow to the Bi-LSTM block to make the data flow clearer, the revised Figure 1B is show as below:

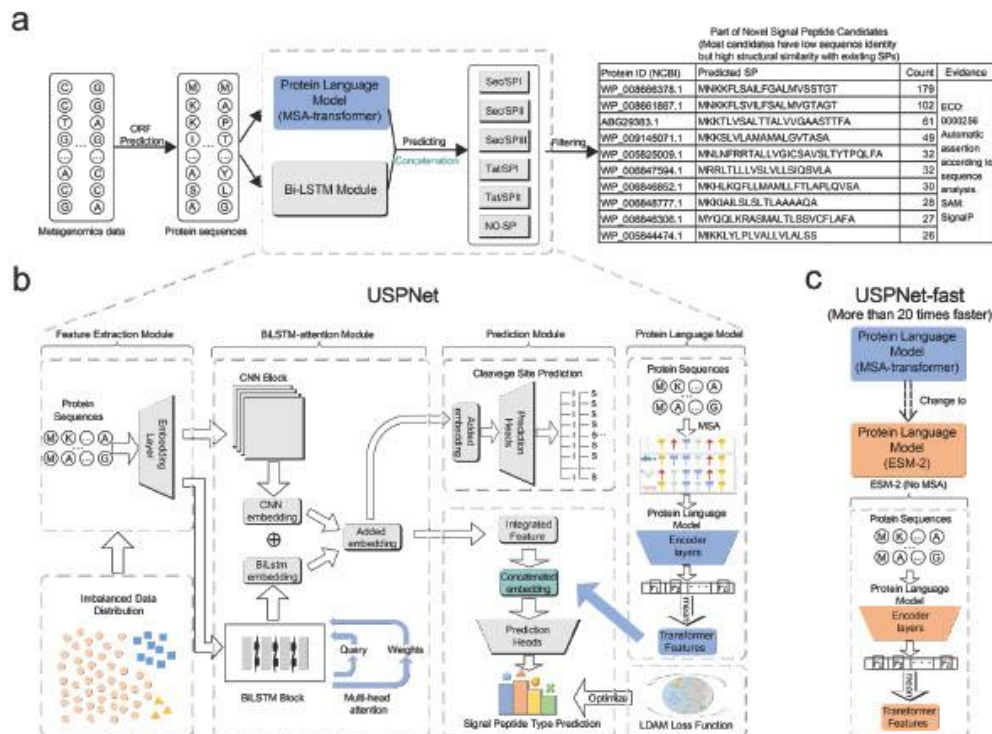


Figure 1.b: Detailed architecture of USPNet. The training data is imbalanced. The protein sequences go through the feature extraction module and are then passed to the BiLSTM module, which includes a Bi-LSTM layer with self-attention and a CNN for extracting long-distance dependencies and features of the sequences. For SP type prediction, USPNet incorporates MSA embeddings generated by a pre-trained MSA Transformer model. Subsequent MLP-based modules predict cleavage sites and signal peptide types. Label Distribution-Aware Margin (LDAM) loss is employed in training to address data imbalance.

* Data in the radar plots of 2d would be appreciated as tables also. The data in Figure 2e would also be appreciated in a supplemental table (apologies if I've missed them).

Answer: Thanks for this comment. In the original version of our manuscript, we have provided the results of Figure 2d in the tabular form, you could find them in Supplementary Table 1-4. And Figure 2e shows the head-to-head comparison between our methods and the SOTA method: SignalP6.0, the experiments are also on the benchmark set. For the signal peptide type prediction, results can also be retrieved from Supplementary Table 1-4. For the cleavage site prediction, the results can be found from Supplementary Table 15-18. And for your convenience, we list some tables below for your reference:

Supplementary Table 1: Benchmarking of Sec/SPI signal peptide detection predictions

Method	Archaea		Eukaryotes	Gram-negative		Gram-positive	
	MCC1	MCC2	MCC	MCC1	MCC2	MCC1	MCC2
SignalP6.0	0.757	0.767	0.799	0.846	0.668	0.913	0.774
DEEPSIG[1]	n.d.	n.d.	0.77	0.718	0.243	0.827	0.234
LipoP[2]	0.698	0.534	0.323	0.759	0.594	0.913	0.574
PRED-LIPO[3]	0.673	0.515	0.167	0.689	0.433	0.913	0.574
PRED-SIGNAL[4]	0.904	0.662	0.236	0.66	0.2	0.855	0.285
PRED-TAT[5]	0.767	0.652	0.34	0.722	0.299	0.825	0.329
TOPCONS2[6]	0.612	0.352	0.49	0.838	0.28	0.82	0.177
USPNet-fast	0.729	0.74	0.854	0.822	0.75	0.956	0.96
USPNet	0.762	0.771	0.876	0.863	0.811	1.0	1.0

Supplementary Table 15: Benchmark results for cleavage site predictions in Sec/SPI signal peptide

Method	Archaea		Eukaryotes		Gram-negative		Gram-positive	
	CS recall	CS precision	CS recall	CS precision	CS recall	CS precision	CS recall	CS precision
SignalP6.0	0.500	0.643	0.747	0.661	0.639	0.534	0.800	0.632
USPNet-fast	0.556	0.377	0.705	0.460	0.705	0.494	0.933	0.636
USPNet	0.528	0.442	0.692	0.605	0.721	0.629	0.933	0.824

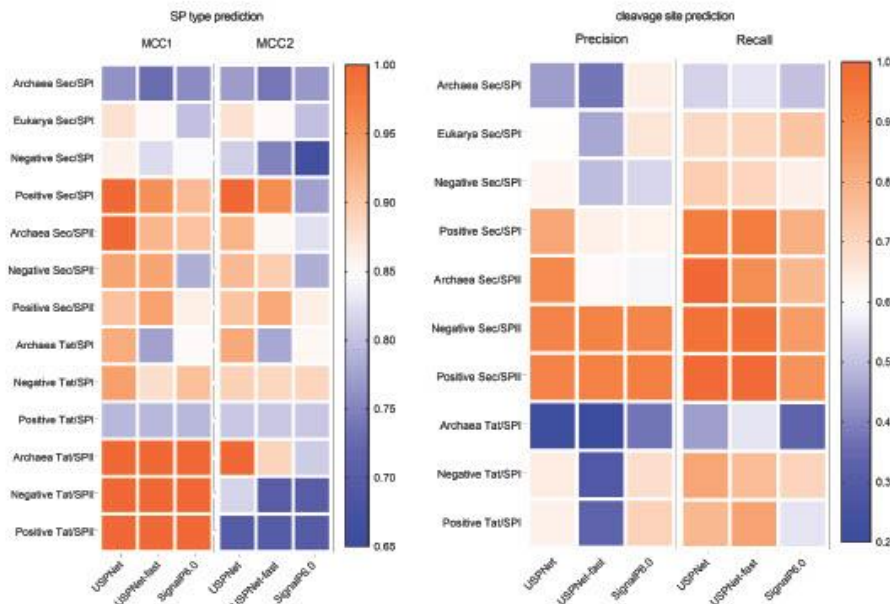


Figure 2.e: Comparison between USPNet, USPNet-fast, and SignalP6.0 on both signal peptide type prediction and signal peptide cleavage site prediction.

* Appendix line 255: Include which C hyperparameters you've tried and which one was best.

Answer: Thanks for this comment. The class weights we have tried in SP type classification include 1) identical weight for all class; 2) inverse of sample frequencies for each class. And for the cleavage site prediction, we have tried different values (1,2,3,4,5,6,7,8,9) and found that set 6 to the class signifying the cleavage site, while set 1 to the other classes would make the performance higher.

Therefore, to clarify the class weight we used in training, we added the best weights used for training in Section 4.2 Training details. The added content is as shown below:

Current version: We use LDAM loss as the objective function in training. For the signal peptide prediction objective function, L_s , class-balanced re-weighting is utilized to assign weights based on the inverse of sample frequencies for each class. In the cleavage site prediction objective function, L_c , a heightened weight of 6 is attributed to the class signifying the cleavage site, the class representing paddings is assigned with weight of 0, while a standard weight of 1 is assigned to all other classes.

And for the C hyperparameter in the margin item of the LDAM loss, we actually applied the value introduced in the paper that developed the LDAM loss (Cao, Wei et al. 2019). In our original manuscript, we said that ' C is a hyper-parameter to be tuned'. And this kind of presentation may lead to the misunderstanding that we have tried different C value. Therefore, in the revised manuscript, we rewrite it and show the exact C we used in line 604.

C value:

$$\text{We adopt } C = \frac{0.5}{\max_{1 \leq j \leq K} \left(n_j^{-\frac{1}{4}} \right)}.$$

* Making the datasets available at osf.io instead of as google drive links.

Answer: Thank you for the suggestion! We found it is very helpful and convenient to use OSF to manage our data. Hence, we have created a OSF project to store the data of our project, we also provided the [link of our project](#) in our github repo for use. Afterwards, we will also continue to update our project on both github and OSF.

Reviewer #4 (Remarks on code availability):

I have reviewed but `_not_` installed and run the code.

However, I did a quick skim and everything seems to be in order for reproduction.

References

Cao, K., et al. (2019). "Learning imbalanced datasets with label-distribution-aware margin loss." Advances in neural information processing systems 32.

Gibson, S., et al. (2017). "N-terminal or signal peptide sequence engineering prevents truncation of human monoclonal antibody light chains." Biotechnology and Bioengineering 114(9): 1970-1977.

Rao, R. M., et al. (2021). MSA transformer. International Conference on Machine Learning, PMLR.

Decision Letter, first revision:

Date: 8th November 23 23:56:19

Last Sent: 8th November 23 23:56:19

Triggered By: Jie Pan

From: jie.pan@us.nature.com

To: liyu@cse.cuhk.edu.hk

CC: computacionalscience@nature.com

BCC: jie.pan@us.nature.com

Subject: AIP Decision on Manuscript NATCOMPUTSCI-23-0326B

Message: Our ref: NATCOMPUTSCI-23-0326B

8th November 2023

Dear Dr. Li,

Thank you for submitting your revised manuscript "USPNet: unbiased organism-agnostic and highly sensitive signal peptide predictor with deep protein language model" (NATCOMPUTSCI-23-0326B). It has now been seen by the original referees and their comments are below. The reviewers find that the paper has improved in revision, and therefore we'll be happy in principle to publish it in Nature Computational Science, pending minor revisions to satisfy the referees' final requests and to comply with our editorial and formatting guidelines.

We are now performing detailed checks on your paper and will send you a checklist detailing our editorial and formatting requirements in about a week. Please do not upload the final materials and make any revisions until you receive this additional information from us.

TRANSPARENT PEER REVIEW

Nature Computational Science offers a transparent peer review option for original research manuscripts. We encourage increased transparency in peer review by publishing the reviewer comments, author rebuttal letters and editorial decision letters if the authors agree. Such peer review material is made available as a supplementary peer review file. **Please remember to choose, using the manuscript system, whether or not you want to participate in transparent peer review.**

Please note: we allow redactions to authors' rebuttal and reviewer comments in the interest of confidentiality. If you are concerned about the release of confidential data, please let us know specifically what information you would like to have removed. Please note that we cannot incorporate redactions for any other reasons. Reviewer names will be published in the peer review files if the reviewer signed the comments to authors, or if reviewers explicitly agree to release their name. For more information, please refer to our [FAQ page](https://www.nature.com/documents/nr-transparent-peer-review.pdf).

Thank you again for your interest in Nature Computational Science. Please do not hesitate to contact me if you have any questions.

Sincerely,

Jie Pan, Ph.D.
Senior Editor
Nature Computational Science

ORCID

IMPORTANT: Non-corresponding authors do not have to link their ORCIDs but are encouraged to do so. Please note that it will not be possible to add/modify ORCIDs at proof. Thus, please let your co-authors know that if they wish to have their ORCID added to the paper they must follow the procedure described in the following link prior to acceptance: <https://www.springernature.com/gp/researchers/orcid/orcid-for-nature-research>

Reviewer #1 (Remarks to the Author):

The revision sufficiently addressed my previous concerns

Reviewer #2 (Remarks to the Author):

The authors addressed my concerns and now the present version is more satisfactory

Reviewer #3 (Remarks to the Author):

The authors have effectively addressed all previously raised concerns, and I believe the manuscript is now suitable for publication.

Reviewer #4 (Remarks to the Author):

Overall I find the article significantly improved! I particularly liked the updated evaluation on the 40% cutoff sets.

All of my major suggestions (extra ablations with the standard cross-entropy objective to support the LDAM loss; comparing against random protein embeddings to study the effect of MSA and ESM embeddings) were explored, and with interesting (at least to me) results that are also included in the manuscript. Thanks!

Additionally, my minor suggestions of extra clarifications in a figure and appendices (tables for the radar plots; and a discussion of the C hyperparameter) have also been included.

Final Decision Letter:

Date: 22nd November 23 10:23:08
Last Sent: 22nd November 23 10:23:08
Triggered By: Jie Pan
From: jie.pan@us.nature.com
To: liyu@cse.cuhk.edu.hk
CC: computacionalscience@nature.com
BCC: fernando.chirigati@us.nature.com
Subject: Decision on Nature Computational Science submission NATCOMPUTSCI-23-0326C
Message: 22nd November 2023

Dear Dr. Li,

I am delighted to tell you that your manuscript NATCOMPUTSCI-23-0326C has been accepted for publication in Nature Computational Science.

As discussed, we will publish your paper on an accelerated schedule. **Please carefully review the details below and contact us immediately at computacionalscience@nature.com if you have any travel plans or other conflicts that may make you unable to respond to us for the next 5-7 days.**

In approximately 2 business days you will receive a link to choose the appropriate publishing options for your paper and complete the appropriate grant of rights necessary to publish your work. As it is vital that this process not be delayed, we strongly encourage you to [whitelist](https://www.simpleminds.com/how-to-check-your-spam-filter-and-whitelist-emails/) the email address do-not-reply@springernature.com to ensure that this message is received.

You will receive a link to your electronic proof via email with a request to make any necessary corrections as soon as possible. You will find that we have made minor changes to enhance the clarity of the text and to ensure that your paper conforms to the journal's style so we ask that you review these proofs carefully to ensure that we have not inadvertently introduced errors or altered the sense of your text in any way.

Please return your proof within 24 hours of receiving it. If you have any questions about your proofs or anticipate any delays please contact rjsproduction@springernature.com immediately.

Once a publication date is set for your paper, the Springer Nature press office will be in touch with the full embargo details. We request that you do not send out your own publicity or contact any journalists until you hear from us that the paper has a confirmed publication date.

If you would like to inform your Public Relations or Press Office about your paper, we

suggest that you do so immediately to allow them as much time as possible to prepare an appropriate press release and organize publicity if they choose to do so. Please include your manuscript tracking number NATCOMPUTSCI-23-0326C and the name of the journal, which they will need if they contact our press office.

Please note that Nature Computational Science is a Transformative Journal (TJ). Authors may publish their research with us through the traditional subscription access route or make their paper immediately open access through payment of an article-processing charge (APC). Authors will not be required to make a final decision about access to their article until it has been accepted. [Find out more about Transformative Journals](https://www.springernature.com/gp/open-research/transformative-journals)

Authors may need to take specific actions to achieve [compliance with funder and institutional open access mandates](https://www.springernature.com/gp/open-research/funding/policy-compliance-faqs). If your research is supported by a funder that requires immediate open access (e.g. according to [Plan S principles](https://www.springernature.com/gp/open-research/plan-s-compliance)) then you should select the gold OA route, and we will direct you to the compliant route where possible. For authors selecting the subscription publication route, the journal's standard licensing terms will need to be accepted, including [self-archiving policies](https://www.springernature.com/gp/open-research/policies/journal-policies). Those licensing terms will supersede any other terms that the author or any third party may assert apply to any version of the manuscript.

If you have any questions about our publishing options, costs, Open Access requirements, or our legal forms, please contact ASJournals@springernature.com.

If you have not already done so, we strongly recommend that you upload the step-by-step protocols used in this manuscript to the Protocol Exchange. Protocol Exchange is an open online resource that allows researchers to share their detailed experimental know-how. All uploaded protocols are made freely available, assigned DOIs for ease of citation and fully searchable through nature.com. Protocols can be linked to any publications in which they are used and will be linked to from your article. You can also establish a dedicated page to collect all your lab Protocols. By uploading your Protocols to Protocol Exchange, you are enabling researchers to more readily reproduce or adapt the methodology you use, as well as increasing the visibility of your protocols and papers. Upload your Protocols at www.nature.com/protocolexchange/. Further information can be found at www.nature.com/protocolexchange/about.

You can now use a single sign-on for all your accounts, view the status of all your manuscript submissions and reviews, access usage statistics for your published articles and download a record of your refereeing activity for the Nature journals.

An online order form for reprints of your paper is available at <https://www.nature.com/reprints/author-reprints.html>. All co-authors, authors' institutions and authors' funding agencies can order reprints using the form appropriate to their geographical region.

Sincerely,

Jie Pan, Ph.D.
Senior Editor
Nature Computational Science

P.S. Click here if you would like to recommend Nature Computational Science to your librarian - this will link directly to the Recommend page.

<http://www.nature.com/subscriptions/recommend.html#forms>

** Visit the Springer Nature Editorial and Publishing website at www.springernature.com/editorial-and-publishing-jobs for more information about our career opportunities. If you have any questions please click here.**