nature portfolio

Peer Review File

Π-PrimeNovo: An Accurate and Efficient Non-Autoregressive Deep Learning Model for De Novo Peptide Sequencing

Corresponding Author: Professor Siqi Sun

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
Unlike conventional methods, the proposed π-PrimeNovo tried to address the limitations of autoregressive models and sequential decoding algorithms, demonstrating significant improvements in sequencing accuracy and computational efficiency. These advancements make it a promising tool for large-scale peptide sequencing applications.

However, several critical areas require further clarification:

1. Data and Methodology:
* A detailed description of the training and testing dataset generation processes is essential for reproducibility and a comprehensive understanding of the evaluation.
The comparison with GraphNovo and PepNet would benefit from a consistent experimental setup, including training on the MassIVE-KB dataset for all methods.
* A clear explanation of the non-autoregressive approach, including its differentiation from the autoregressive counterpart, is necessary. Moreover, crucial model parameters such as the number of tokens, maximum peptide length, and training termination criteria should be explicitly stated.

2. Performance Evaluation:
* While the reported high recall is commendable, incorporating additional metrics like precision would provide a more comprehensive and comparable evaluation, aligning with common practices in de novo sequencing benchmarking.
* The visualization in Figure 3g could be enhanced with a Venn diagram to offer clearer insights into the generalization capabilities across different datasets.

3. Experimental Details and Reproducibility:
* The typo in line 317 should be corrected.
* Detailed information about the database search software, configuration settings, and relevant references should be provided for the quality control process.
* A clear description of any pre-processing steps applied to the MassIVE-KB dataset is essential.
* Uploading the trained model and all associated results would significantly enhance reproducibility.
* The provided code should be thoroughly checked for errors and dependencies to ensure its functionality.

(Remarks on code availability)
1. No trained models and related results for PrimeNovo were provided.
2. The provided code cannot be executed without the correct model file. Additionally, there are bugs in the code, such as `ModuleNotFoundError: No module named 'PrimeNovo.components.encoders'`. Please ensure all code is functional and error-free.

Reviewer #2

(Remarks to the Author)

The manuscript described a new AI based de novo peptide sequencing tool Pi-PrimeNovo. The main novelty compared to other similar tools is the use of a non-autoregressive model. In addition, it uses a precise mass control to ensure the generated peptide matches the precursor mass of the spectrum. There are other attempt to match the precursor match in AI-based de novo sequencing, but the method used here is new.

The model has been extensively benchmarked on large mass spectrometry datasets. The performance of PrimeNovo exceeds the state of the art significantly in terms of both de novo sequencing accuracy and speed. It also demonstrates excellent accuracy in detecting PTMs.

The main contribution of the paper is the software. The availability of the software will be a great addition to the proteomics field. In terms of the software availability, I do not see the manuscript discuss the availability of the software. The code and software submission form claimed the software is MIT license. The code is also attached in the reviewer's package. However, I do not see the model's weights file. The model cannot run without the weights file. The training of the weights is a critical part of a neural network model. The reusability of their results is severely limited if the weights were not to be published.

Besides this main concern of software availability, I only have the following minor suggestions:

1. p3. line 68-70: Should the "all spectra" be "all assigned spectra"? If a spectrum is not identified with database search, it is hard to know ground-truth sequence for the spectrum and is often excluded from de novo sequencing benchmark.

2. Since speed is an important part of the contribution, provide the de novo sequencing speed in absolute terms (in addition to the ratio to Casanovo).

3. Figure 3c and 3g. Provide a little more explanation about the figure.

4. p10, line 315: intermediate fragment ions -> internal fragment ions

5. p10. line 351: "rigorous quality control process T\U\D\DS"

What is T\U\D\DS?

7. Supplementary information, section 2.1 CTC loss calculation with dynamic programming.

Will the use of this dynamic programming based loss function affect the back-propagation algorithm for training the neural network? Please provide a brief explanation.

(Remarks on code availability)
I only checked the overall availability of different components of the code. But I didn't not review every line.

Reviewer #3

(Remarks to the Author)
In this paper, the authors present a new transformer architecture for peptide De novo sequencing from MS/MS data. The new architecture together with CTC loss functon(which is often used in the voice recognition task) seems applied to the peptide De novo sequencing task successfully. The non-autoregressive model architecture gains significant performance improvement over the autoregressive models like Casanovo. The authors tested their model on some previously published benchmark datasets together with some other datasets for special applications. Different metrics are used for all the dataset. The results seem convincing. Generally I recommend publication for this paper. However, there are several major issues requiring attention before the publication:
1. Although source code is provided for the review, the models are not provided which makes it hard to reproduce their results. If models are not provided, at least some description should be included for the training process and hyper parameter settings.
2. Peptide recall is used for most data set evaluation. However, amino acid recall on all identified psms is a more robust metric to avoid the overfitting problem for the deep neural network models. It should be also included
3. For the GraphNovo and PepNet, to make it a fair comparison, better use the same training data for different models.
4. For speed comparison with Casanovo V2, the Casanovo speed seems quite inconsistent with our own experience. The authors better contact Casanovo authors to make sure their parameter settings are optimised for Casanovo.
5. All the data in this paper are acquired through the Thermo orbitrap instrument. It needs to include the data from other instruments, e.g.,TimsTOF, and with some comparison and discussion for the performance.

(Remarks on code availability)
Both training and inference code are provided. But no detail description for training process. Also no model is provided which makes it hard to run the code.

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
I appreciate that the authors have addressed some of my previous comments; however, the following concerns still need further explanation:
1. Line 364: The manuscript states, "As depicted in Fig. 5f, proteins identified by PrimeNovo and Casanovo were correctly assigned to 10 genera, 14 species, and 20 COG (Clusters of Orthologous Groups of proteins) categories." However, there is no Fig. 5f in the manuscript. Please clarify the correct figure reference.
2. Peptide-to-Protein Mapping: The performance of the proposed models was evaluated at the protein level, but the process of mapping peptides to proteins is unclear. Could you provide more details on the methodology used for this mapping? Were third-party tools employed, or was an in-house script developed for this purpose? Additionally, what criteria were used to map peptides to proteins?
3. Peptides Not Mapped to Proteins: Among the peptides identified by the proposed models, how many were not mapped to any proteins? A discussion on these unmapped peptides would be beneficial. Were they potentially novel peptides with post-translational modifications (PTMs) or protein variants? Alternatively, could they be peptides with amino acid prediction errors?
4. Fig. 5b: The peptides under "PrimeNovo" in Fig. 5b—do these numbers represent peptides that could not be mapped to any bacteria? If so, an explanation of these peptides would be helpful.
5. Comparison to Database Searching Tools: Is there any discussion comparing the proposed method to traditional database searching tools? Can the proposed model replace conventional database searching, or does it serve a complementary role?
6. Line 418: The manuscript reports that the classification accuracies for all PTMs exceeded 95%, except for asymmetric and symmetric dimethylation at Arginine (R) and monomethylation at Arginine (R), which have accuracies of 77%, 77%, and 69%, respectively. Could the authors provide an explanation for why these specific PTMs exhibit lower accuracies compared to others? Furthermore, given that recall rates are comparable to other datasets without special PTMs, does this suggest that fine-tuning on PTM-enriched datasets may not be necessary?
7. Fine-Tuning for Phosphorylation Dataset: The manuscript mentions that PrimeNovo was fine-tuned on a training dataset and tested on phosphorylation data from Xu et al., achieving a classification accuracy of 98% and a peptide recall rate of 66%. Why was additional fine-tuning applied specifically for the phosphorylation dataset, given that the model had already been fine-tuned on the 21 PTMs dataset? What would the performance have been on the phosphorylation dataset without this additional fine-tuning? Is fine-tuning always required in practice when applying the model to PTM analysis?
8. Hyperparameters: How many hyperparameters were involved in the proposed model? A brief discussion on hyperparameter selection and its impact on the model's performance would be informative.
9. Line 199: The manuscript states that PrimeNovo had 34,747 overlapping PSMs with MaxQuant search results, while Casanovo V2 and Casanovo had 26,591 and 16,814, respectively. It is encouraging to see that PrimeNovo demonstrates more consistency in its predictions compared to traditional database-searching tools. However, it would be useful to know how many peptides were reported by traditional database-search tools but missed by the de novo sequencing methods. Additionally, how many amino acid mismatches were observed when comparing peptides identified by de novo sequencing to those identified by traditional methods? Discussing the potential limitations of the proposed model and de novo sequencing would provide valuable context.

(Remarks on code availability)
The proposed code could be run on the provided sample data.

Reviewer #2

(Remarks to the Author)
The revised manuscript sufficiently addressed all my earlier concerns.

(Remarks on code availability)

Reviewer #3

(Remarks to the Author)
The author has addressed all my concerns in my previous review. However I just noticed there is still a small error on page 15, below line 518, the formula for spectrum encoding might still be wrong. The authors can reference paper https://www.nature.com/articles/s41467-024-49731-x for the correct formula.

(Remarks on code availability)
Try run the code with the model provided. It can finish successfully and give the meaningful results.

Response to Reviewers

Manuscript ID: NCOMMS-24-37565

Manuscript title: π-PrimeNovo: An Accurate and Efficient Non-Autoregressive Deep Learning Model for De Novo Peptide Sequencing

We sincerely thank all the reviewers for their time and effort in reviewing our manuscript and for providing valuable feedback. We especially appreciate the reviewers for acknowledging the effectiveness of PrimeNovo and recognizing the novelty of our proposed algorithm. We believe that the comments and suggestions provided have significantly contributed to improving our manuscript, and we have addressed all the concerns with corresponding revisions in the paper.

We first highlight our response to some of the major concerns raised by the reviewers:

1. **Code Availability and Usability:** We have re-uploaded the compressed zip file containing all code, along with detailed instructions for running PrimeNovo. After double-checking, we can confirm the code is correct and should run smoothly by following the instructions in our `README.md` file. The model weights are also linked in the `README`, available for download from Google Drive to replicate our results. We include most important model configuration and running instruction in our `README.md` here:

   `Model Settings:`

   - **n_beam**: number of CTC-paths (beams) considered during inference. We recommend a value of 40. Considering all decoding paths lead to slower inference speed and considering too few paths lead to lower performance.
   - **mass_control_tol**: This setting is only useful when **PMC_enable** is `True`. The tolerance of PMC-decoded mass from the measured mass by MS, when the mass control algorithm (PMC) is used. For example, if this is set to 0.1, we will only obtain peptides that fall under the mass range [measured_mass-0.1, measured_mass+0.1]. `Measured mass` is calculated by: (*pepMass* - 1.007276) * *charge* - 18.01. *pepMass* and *charge* are given by the input spectrum file (MGF).
   - **PMC_enable**: Whether to use the PMC decoding unit or not, either `True` or `False`. If disabled, it will use naive CTC decoding with `n_beam`.
   - **n_peaks**: Number of the most intense peaks to retain; any remaining peaks are discarded. We recommend a value of 800.
   - **min_mz**: Minimum peak m/z allowed; peaks with smaller m/z are discarded. We recommend a value of 1.
   - **max_mz**: Maximum peak m/z allowed; peaks with larger m/z are discarded. We recommend a value of 6500.
   - **min_intensity**: Minimum peak intensity allowed; less intense peaks are discarded. We recommend a value of 0.0.

   `Run PrimeNovo:`

   First download the following files for evaluation on the given test MGF (you can replace our MGF with your own MGF file):

   - `model_massive.ckpt`: https://drive.google.com/file/d/12IZgeGP3ae3KksI5_82yuSTbk_M9sKNY/view?usp=share_link
   - `Bacillus.10k.mgf`: https://drive.google.com/file/d/1HqfCETZLV9ZB-byU0pqNNRXbaPbTAceT/view?usp=drive_link

Then run:

```
python -m PrimeNovo.PrimeNovo --mode=eval --peak_path=./bacillus.10k.
    mgf --model=./model_massive.ckpt
```

Upon acceptance of the paper, we would commit to releasing the full codebase and model weights on GitHub for public use. Additionally, we plan to enhance the software's usability by integrating the model into a Python library, providing one-click installation and easy usage. This feature will be announced on our GitHub page once completed, and we will continue to maintain and update the repository with new releases.

2. **Performance Comparison with Recent Methods (GraphNovo and PepNet):** To ensure a fair comparison with methods like GraphNovo and PepNet, which were trained on different datasets than PrimeNovo, we have re-trained PrimeNovo using exact same training dataset as GraphNovo and PepNet, respectively. Our results, presented in Rebuttal Figure 1 and Rebuttal Figure 2, show that PrimeNovo consistently outperforms both GraphNovo and PepNet on their respective test sets. Specifically, when trained on the GraphNovo training dataset, PrimeNovo improves GraphNovo's 66% average peptide recall to 72% across the 3-species test set used by GraphNovo. Similarly, when trained on the PepNet training dataset, PrimeNovo improves PepNet's 66% average peptide recall to 72% on PepNet's testing data. These results further highlight PrimeNovo's superior performance under equitable conditions.

3. **Speed Comparison with Casanovo V2:** After extensive investigation and additional experiments, we discovered that inference batch size significantly impacts speed due to the effect on GPU and CPU offloading and queuing. We identified the optimal batch size for both PrimeNovo and Casanovo V2, accounting for the use of post-decoding strategies (PMC and beam search). The updated speed comparison in our manuscript reflects that PrimeNovo demonstrates even faster inference times when using the optimal batch size.

We also summarize some of the key revisions made to our main paper and Supplementary Information:

1. We have provided more detailed information on all datasets used, including their sources, processing steps, and splitting strategies, in the Supplementary Information.
2. In addition to the Amino Acid (AA) precision results included in the main paper, we now also report AA-level recall results for the main benchmark test data and analyze the differences in these two metrics.
3. We have conducted experiments applying PrimeNovo to data acquired from instruments other than the Thermo Orbitrap, specifically timsTOF Pro. We analyze the model's generalizability and performance differences compared to previous state-of-the-art methods.
4. We have added further explanations regarding the model's architecture specifications, as well as clarified the differences between our non-autoregressive design and previous autoregressive models.

Please find below for our point-by-point detailed response to each reviewer's comments:

- Blue: Reviewer's comments.
- Black: Our response.
- Orange: Reference to the (updated) manuscript.

Reviewer 1:

Unlike conventional methods, the proposed π-PrimeNovo tried to address the limitations of autoregressive models and sequential decoding algorithms, demonstrating significant improvements in sequencing accuracy and computational efficiency. These advancements make it a promising tool for large-scale peptide sequencing applications.

Re: We thank the reviewer for recognizing our contribution in the proposed method and for highlighting the great potential of PrimeNovo in large-scale proteomics applications. We have conducted additional experiments and provided further explanations for some unclear aspects of our algorithms. Below, we include our point-by-point responses to your questions:

1. Data and Methodology:

1.1 A detailed description of the training and testing dataset generation processes is essential for reproducibility and a comprehensive understanding of the evaluation.

Re: We thank the reviewer for the suggestions regarding the explanation of the data generation process, as it is indeed one of the most important aspects when it comes to training and evaluating our model. We employed the Massive-KB dataset for training PrimeNovo, while employing diverse datasets (nine-species, PT, 21PTMs, PXD019483, HCC, 2020-Cell-LUAD, IgG1-Human-HC, three-species, revised nine-species, and Cell-metaproteome) for testing purposes. Except for the nine-species benchmark dataset and PepNet training dataset, all other datasets were downloaded as raw files along with identification results. For these datasets, we first converted the raw files into mgf files by MSConvert (version 3.0), then incorporated the peptide sequences, charge states, and precursor ion masses from the identification results into these mgf files. For datasets where the distinction between training and testing sets was not made during download, we devised a new strategy for partitioning into training and testing sets. Specifically, we randomly selected a certain number of PSMs from the dataset for testing, while reserving the remaining annotated data for training and fine-tuning. To ensure that none of the true peptides in the testing set appear in the training set, we implemented a selection method where the inclusion of any PSM in the test set automatically results in the selection of all other PSMs with the same true peptide sequence to the test set. This prevents any leakage of answers during fine-tuning/training and guarantees a fair testing environment. The detailed generation processes of the training and testing dataset are as follows:

- For the MassIVE-KB dataset [1], the raw files (in mzML format) and the filtered identification results from the “All Candidate library spectra” section of the MassIVE Knowledge Base spectral library v1 were obtained by downloading from the MassIVE repository. This dataset contains around 30 million PSMs, which were all used as training set.
- For the nine-species benchmark dataset [2], the mgf files were directly downloaded from the MassIVE repository (identifier: MSV000081382), shared by the authors of the DeepNovo paper. We did not undertaken any re-partitioning for this dataset and used downloaded test set directly for testing our model.
- For the PT [3] dataset, the raw files and MaxQuant identification results were obtained by downloading from the PRIDE [4] repository by PXD004732. We applied the training and testing splitting strategy as described above, selecting 58,000 PSMs as the test set and using the remaining data for training and fine-tuning.
- For the 21PTMs [5] dataset, the raw files and MaxQuant identification results were obtained by downloading from the PRIDE [4] repository by PXD009449. This dataset consists of 24 portions, with 21 portions corresponding to different post-translational modifications (PTMs), each annotated with a specific PTM, and 3 portions containing non-PTM data. Following our splitting strategy, we split all 24 portions using a 95% training to 5% testing ratio for training and fine-tuning. Each PTM portion was then combined with its corresponding non-PTM portion. For example, the Trimethyl of K amino acid training data was combined with Non-PTM of K amino acid training data to form the training set for Trimethyl of K amino acid. Similarly, Trimethyl of K amino acid testing data was combined with Non-PTM of K amino acid testing data to evaluate the accuracy of PTM identification.

- For the PXD019483 [6] dataset, the raw files and MaxQuant identification results were obtained by downloading from the PRIDE [4] repository by PXD019483. This dataset was used as the test dataset for PepNet [7]. We did not undertaken any re-partitioning for this dataset and used downloaded test set directly for testing our model.

- For the PepNet training dataset [7]: The mgf files were provided by the authors [7] after we contacted them. We downloaded their training and testing data and made no further changes or splitting before using them for training/fine-tuning and testing.

- For the HCC [8], the raw files and MaxQuant identification results were obtained by downloading from the iProX [9] repository (identifier: IPX0000937000). We randomly selected 56,000 PSMs from the HCC dataset for testing, following our proposed splitting strategy. Since our fine-tuning on the HCC dataset was performed using varying sizes of training data (Figure 3.e in Main Manuscript), we randomly selected 100, 1,000, 10,000, and 100,000 PSMs from the entire splitted training set, all of which have no overlap with the testing set used here.

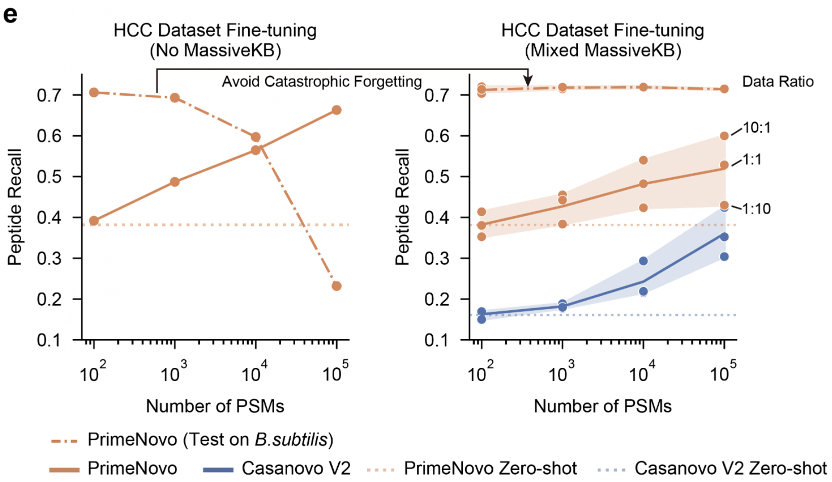


Figure 3.e in Main Manuscript

- For the 2020-Cell-LUAD [10] datasets, the raw files and MaxQuant identification results were obtained by downloading from the iProX [9] repository (identifier: IPX0001804000). We applied the training and testing splitting strategy as described above, selecting 58,000 PSMs as the test set and using the remaining data for training and fine-tuning.

- For the IgG1-Human-HC [11] dataset, the raw files and the combined identification results of the database algorithms MS-GF+ and X!Tandem were obtained by downloading from the MassIVE repository (identifier: MSV000079801). This dataset was compiled by researchers [11] to comprehensively evaluate de novo peptide sequencing methods. We applied it directly for our testing with no splitting.

- For the three-species dataset and GraphNovo training data [12], the mgf files and the SEQUEST search results were obtained by downloading from the data shared by the GraphNovo authors on Zenodo (identifier: zenodo.8000316). The authors of GraphNovo [12] provided its own training data along with three different testing species. We used its splitted training set directly for fine-tuning/training and splitted testing portion for testing our model without any modifications.

- For the revised nine-species benchmark dataset [13], the raw files and Crux identification results were obtained by downloading from the MassIVE repository (identifier: MSV000090982). This dataset was proposed by Casanovo V2 [13] for evaluating and testing de novo peptide sequencing models. We downloaded the processed data and used it directly for testing our model without any data splitting.

- For the Cell-metaproteome dataset [14], the raw files and MyriMatch identification results were obtained by downloading from the MassIVE repository (identifier: MSV000082287).

Based on the reviewer's suggestions, we have made the following modifications to the 'Datasets' section in our Supplementary Information as follows:

Name	Species	Instrument	Accession	No. of PSMs	Description
nine-species benchmark	*V. mungo* *M. musculus* *M. mazei* *B. subtilis* *C. endoloripes* *S. lycopersicum* *S. cerevisiae* *A. mellifera* *H. sapiens*	Q-Exactive	MSV000081382	37,775 37,021 164,421 291,783 150,611 290,050 111,312 314,571 130,583	A standard benchmark previously employed. Analysis conducted through leave-one-species-out cross-validation or a zero-shot approach
MassIVE-KB	*H. sapiens*	Q-Exactive	-	30,633,841	Foundational dataset for the development of Casanovo V2 and PrimeNovo algorithms
Proteometools (PT)	*H. sapiens*	Orbitrap Fusion Lumos	PXD004732	28,065,572	Employed as test datasets for PrimeNovo and baseline approaches, analyzed under zero-shot or few-shot fine-tuning scenarios
HCC	*H. sapiens*	Orbitrap Fusion Lumos	IPX0000937000	20,217,331	
IgG1-Human-HC	*H. sapiens*	LTQ Orbitrap	MSV000079801	14,087	
three-species	*A. thaliana* *C. elegans* *E. coli*	Orbitrap Q Exactive HF-X QExactive HF Q-Exactive	zenodo.8000316	12,222 12,103 12,330	
PXD019483	*H. sapiens*	Q Exactive HFX Orbitrap	PXD019483	1,915,890	
revised nine-species benchmark	nine-species	Q-Exactive	MSV000090982	2,812,194	
PepNet training dataset	multiple	multiple	zenodo.13352403	3,041,555	

Supplementary Table 1: Summary of training and testing datasets in our study.

Supplementary Table 1 provides a comprehensive overview of the training and testing datasets utilized in our study. This includes the Massive-KB dataset, which was used for training PrimeNovo, and the nine-species benchmark dataset, employed for training PrimeNovo CV and for conducting comparative evaluations against baseline models. Following the training phase, PrimeNovo was subsequently evaluated on a diverse set of datasets, encompassing PT, HCC, IgG1-human-HC, three-species, PXD019483, and the revised nine-species benchmark datasets. Except for the nine-species benchmark dataset and PepNet training dataset, all other datasets were downloaded as raw files along with identification results. For these datasets, we first converted the raw files into mgf files by MSConvert (version 3.0), then incorporated the peptide sequences, charge states, and precursor ion masses from the identification results into these mgf files. For datasets where the distinction between training and testing sets was not made during download, we devised a new strategy for partitioning into training and testing sets. Specifically, we randomly selected a certain number of PSMs from the dataset for testing, while reserving the remaining annotated data for training and fine-tuning. To ensure that none of spectra with the same peptides appear in both the testing set and the training set, we implemented a selection method where the inclusion of any PSM in the test set automatically results in the selection of all other PSMs with the same true peptide sequence to the test set. This prevents any leakage of answers during fine-tuning/training and guarantees a fair testing environment. The detailed generation processes of the training and testing dataset are as follows:

1. **nine-species benchmark** dataset [2]: It was originally curated by Tran *et al.* during the development of DeepNovo [2], comprises a diverse assortment of data from nine distinct species. It was meticulously compiled by aggregating contributions from multiple research teams, with the aim of minimizing biases associated with species and laboratory sources. This dataset has since gained popularity as a preferred choice for evaluating various de novo sequencing algorithms, including Casanovo [13] and

PointNovo [15], utilizing the leave-one-species-out cross-validation method established by DeepNovo. In our study, we accessed this dataset from the MassIVE repository (MSV000090982) and adopted the same cross-validation strategy to ensure a fair and consistent comparison. The mgf files were directly downloaded from the MassIVE repository (identifier: MSV000081382), shared by the authors of the DeepNovo paper. We did not undertaken any re-partitioning for this dataset and used downloaded test set directly for testing our model.

2. **MassIVE-KB** [1]: This dataset played a pivotal role in training our model. It encompasses an extensive collection of over 2.1 million precursors derived from 19,610 proteins. This dataset was meticulously assembled, drawing from more than 31 TB of human data originating from 227 public proteomics datasets. Notably, it upholds rigorous false discovery rate controls, ensuring a comprehensive and robust training environment for our model. The raw files (in mzML format) and the filtered identification results from the "All Candidate library spectra" section of the MassIVE Knowledge Base spectral library v1 (https://massive.ucsd.edu/ProteoSAFe/static/massive-kb-libraries.jsp) were obtained by downloading from the MassIVE repository. We did not split this dataset and used all psms as training set.

3. **ProteomeTools (PT)** [3]: The PT dataset, chosen for PrimeNovo's performance evaluation and fine-tuning, constitutes a carefully curated assembly of synthetic human peptides. As a prominent component of the Proteometools project, it comprises in excess of 330,000 synthetic tryptic peptides, encompassing a diverse spectrum of human gene products. The presence of well-established peptide sequences within this dataset serves as a reliable foundation for evaluating the algorithm's precision and accuracy. The raw files and MaxQuant identification results were obtained by downloading from the PRIDE [4] repository by PXD004732. We applied the training and testing splitting strategy as described above, selecting 58,000 PSMs as the test set and using the remaining data for training and fine-tuning.

4. **HCC** [8]: From a real-world application standpoint, the HCC dataset, centered on proteomics data from early-stage human hepatocellular carcinoma (HCC) patients, was incorporated. This dataset provides a distinctive insight into the proteomics of both tumor and non-tumor tissues, facilitating an evaluation of our model's generalizability in a clinical context.

5. **IgG1-Human-HC** [11]: The IgG1-Human-HC dataset, employed to assess the generalizability of de novo sequencing models, comprises human antibody sequences that have undergone processing with various enzymes, including trypsin, chymotrypsin, and others. This dataset holds particular importance due to its applicability in identifying novel or unfamiliar protein sequences, especially in the realm of immunotherapy antibodies where variable sequences are often unavailable, rendering traditional database search algorithms ineffective. Hence, we utilized this dataset to evaluate our model's proficiency in unraveling the amino acid sequences of unknown antibodies.

6. **three-species** [12]: We employed the three-species dataset, made available by GraphNovo [12], which comprises samples from *Arabidopsis thaliana*, *Caenorhabditis elegans*, and *Escherichia coli*, for our comparative analysis. This dataset, subjected to trypsin digestion and analyzed using state-of-the-art mass spectrometry techniques, allowed us to conduct a thorough performance comparison between PrimeNovo and GraphNovo.

7. **PXD019483** [6]: In order to facilitate a comparison with the recently published PepNet [7], we acquired the publicly available test dataset PXD019483, which is the testing set used by PepNet. This dataset, consisting of extensive human proteomics data, is accessible through the PRIDE repository under the identifier PXD019483. In line with PepNet's approach, we filtered the spectra to include only those with charge states 2+ and 3+, as identified by MaxQuant with a false discovery rate (FDR) threshold of less than or equal to 1%, and with precursor mass differences of no more than 10 ppm. The raw files and MaxQuant identification results were obtained by downloading from the PRIDE [4] repository by PXD019483. This dataset was used as the test dataset for PepNet [7]. We did not undertaken any re-partitioning for this dataset and used downloaded test set directly for testing our model.

8. **Revised nine-species benchmark** [13] : The dataset was acquired by the authors of Casanova V2 [13] through a process that involved downloading the raw files of the nine-species benchmark dataset and subsequently reanalyzing them using Crux version 4.1 [16]. Following this analysis, peptides shared between species were removed. The resulting dataset, known as the new nine-species benchmark, encompasses approximately 2.8 million PSMs and is derived from a total of 343 raw files. For our study,

we directly obtained the identification results that were shared by the authors.

9. **PepNet training dataset** [7]: The mgf files were provided by the authors [7] after we contacted them. We downloaded their training and testing data and made no further changes or splitting before using them for training/fine-tuning and testing.

Name	Species	Instrument	Accession	No. of PSMs	Description
Cell-metaproteome	Microbes	LTQ Orbitrap	MSV000082287	26,457,107	metaproteomics analysis purpose
21PTMs	*H. sapiens*	Orbitrap Fusion Lumos	PXD009449	703,606	PTM analysis purpose
2020-Cell-LUAD	*H. sapiens*	Orbitrap Fusion	IPX0001804000	26,162,410	

Supplementary Table 2: Summary of datasets used in downstream applications.

Supplementary Table 2 presents a comprehensive overview of the datasets utilized for downstream applications in our investigation.

1. **Cell-metaproteome** [14]: This dataset was utilized to assess the performance of our model on complex samples containing multiple coexisting species, such as microbial communities that often lack reference sequences. We accessed a publicly available human-gut-derived bacterial metaproteomics dataset, which was sampled from the human gastric organ and processed using an LTQ Orbitrap mass spectrometer, resulting in the generation of MS/MS spectra. These MS/MS spectra were subjected to analysis using MyriMatch v.2.2. We obtained approximately 26 million PSMs by downloading the identification results generously shared by the authors of the analysis [14]

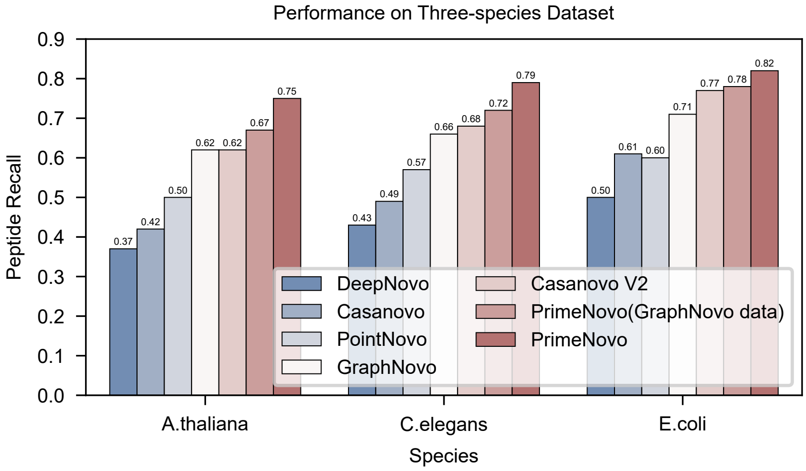
2. **21PTMs** [5]: The 21PTMs dataset represents a publicly available benchmark dataset notable for containing the most extensive variety of post-translational modifications (PTMs) to date. This dataset comprises approximately 5,000 peptides, which in turn represent 21 different naturally occurring human PTMs, encompassing modifications of lysine, arginine, proline, and tyrosine side chains, along with their corresponding unmodified counterparts. the raw files and MaxQuant identification results were obtained by downloading from the PRIDE [4] repository by PXD009449. Following our splitting strategy, we split all 24 portions using a 95% training to 5% testing ratio for training and fine-tuning. Each PTM portion was then combined with its corresponding non-PTM portion. For example, the Trimethyl of K amino acid training data was combined with Non-PTM of K amino acid training data to form the training set for Trimethyl of K amino acid. Similarly, Trimethyl of K amino acid testing data was combined with Non-PTM of K amino acid testing data to evaluate the accuracy of PTM identification.

3. **2020-Cell-LUAD** [10]: To assess our model's ability to infer phosphorylated peptides, we obtained a publicly available phosphoproteomics dataset, known as 2020-Cell-LUAD [10]. This dataset was specifically designed for the study of lung adenocarcinoma (LUAD) and was compiled from the tumors of 103 LUAD patients and their corresponding non-cancerous adjacent tissues. It offers both phospho-enriched data (the phosphoproteomic data) and non-enriched data (the proteomic data), facilitating a comprehensive analysis of phosphorylation events.

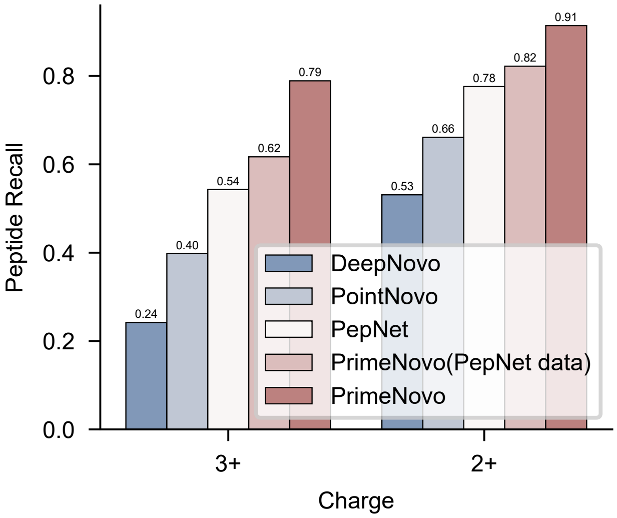
1.2 The comparison with GraphNovo and PepNet would benefit from a consistent experimental setup, including training on the MassIVE-KB dataset for all methods.

Re: We agree with the reviewer that comparisons between models should be conducted using the same training and evaluation data to ensure fairness. For comparisons with previous methods, such as Casanovo V1 [17], V2 [13], DeepNovo [2], and PointNovo [15], we used the same training and testing data and performed the evaluations under fair conditions, as detailed in the manuscript.

However, this is different for GraphNovo and PepNet due to their divergence from the conventional training and testing data (nine-species cross-validation dataset) of all previous methods. As a result, we simply performed evaluations using our trained model on their proposed testing data for simplicity. To further ensure a fair and convincing comparison, we have conducted additional training on PrimeNovo using the training data provided by GraphNovo and PepNet, respectively. The results are demonstrated in Rebuttal Figure 1 and Rebuttal Figure 2. As shown, when trained on the same dataset, PrimeNovo demonstrates superior accuracy compared to GraphNovo, with peptide recall improving from 62% to 67%, 66% to 72%, and 71% to 78% in *A. thaliana*, *C. elegans*, and *E. coli*, respectively (Rebuttal Figure 1). Additionally, our model, trained on the GraphNovo dataset, outperforms Casanovo V2, which was trained on the much larger and more diverse MassiveKB dataset. When trained on the PepNet dataset, PrimeNovo further surpasses PepNet, improving PepNet's 54% peptide call to 62% in peptides with 2+ charges and 78% to 82% in peptides with 3+ charges. These results provide strong evidence of PrimeNovo's superior performance and robust design, which excels in all fair comparisons.



Rebuttal Figure 1: Peformance comparison on 3species testing dataset used by GraphNovo. PrimeNovo (GraphNovo data) is PrimeNovo retrained with GraphNovo training data.



Rebuttal Figure 2: Peformance comparison on test dataset used by PepNet. PrimeNovo (PepNet data) is the PrimeNovo retrained using PepNet training dataset.

Notice that we did not retrain GraphNovo or PepNet on the MassiveKB dataset as required by the reviewer for the following reasons:

1. **Training GraphNovo May Be Resource-Intensive in Terms of GPU Time and Storage:** Based

on our experience, constructing spectrum graphs using GraphNovo's codebase requires substantial storage, reaching terabytes (TB) of disk space for large datasets like MassiveKB. Our replication also indicates that graph construction can be very time-consuming. However, since we did not develop this algorithm, we may not be in the best position to optimize its training efficiency or fully exploit its performance. Consequently, we opted to train PrimeNovo on the same 1.6 million sample dataset used for GraphNovo, enabling a direct comparison with the results reported in the original GraphNovo paper.

2. **Optimal Settings for Both GraphNovo and PepNet Are Difficult to Determine Using MassiveKB:** As is well known, training deep learning models requires extensive parameter tuning to achieve optimal results. In our development, the optimal settings for training PrimeNovo with MassiveKB and training PrimeNovo with nine-species cross-validation differ significantly. Similarly, although GraphNovo and PepNet outline training details for their adopted datasets, achieving optimal performance on the MassiveKB dataset would likely require considerable trial and error to determine the best training settings. Given our limited computational resources, conducting large-scale parameter searches for these two models is challenging. Additionally, since we did not develop these models, we may not be in the best position to achieve their optimal performance on different datasets, as this may require deeper insights from the original authors regarding optimal training strategies and specific parameter settings.

1.3 A clear explanation of the non-autoregressive approach, including its differentiation from the autoregressive counterpart, is necessary. Moreover, crucial model parameters such as the number of tokens, maximum peptide length, and training termination criteria should be explicitly stated.

Re: We thank the reviewer for raising concerns about clarifying the model's detailed configuration and its differences from conventional approaches. To compare autoregressive and non-autoregressive architectures, we have provided several comparisons and explanations in our methodology sections:
The conventional approach to sequence generation involves training a system to predict the probability of the next token, denoted as $P(a_{(i+1)}|a_{1:i})$. This method is known as autoregressive generation, where the likelihood of the $(i+1)$-th amino acid is conditioned on the preceding i amino acids. However, this autoregressive approach imposes a unidirectional flow in the generation process, preventing the system from revising previous outputs. Such a unidirectional flow is at odds with the inherent nature of proteins, where the presence of each amino acid depends on information from both preceding and succeeding amino acids. To overcome this limitation, we propose a non-autoregressive approach to sequence modeling. In this paradigm, all amino acids can be generated simultaneously. This means that each amino acid's generation is not solely reliant on the preceding ones; instead, it can access bidirectional information from the surrounding amino acids. This better aligns with the natural behavior of proteins and significantly enhances the accuracy of our sequence generation system. Formally, in a non-autoregressive system with a predefined maximum generation length t, we model the probability of generating an amino acid, $P(a)$, at each position t independently.

For better clarity, we have further added the following content to the Supplementary Information to better differentiate the architectural difference:
The differences between autoregressive (AR) models and non-autoregressive (NAR) models, specifically in the context of Transformer architecture (AT vs NAT), are detailed as follows, with each counterpart outlined:

1. **Decoder Input and Decoder Masking:** During the training of AT models, all tokens are fed into the Transformer decoder to parallelize the training. However, because the modeling objective for autoregression is the conditional probability $P(a_{i+1}|a_{1:i})$, exposing all tokens to the AR model would lead to information leakage and fail to model the conditional probability correctly. Therefore, AR model training employs "causal masking." This mask forces each attention mechanism to focus only on the preceding tokens by masking the values after each token with $-\infty$, effectively zeroing out those values after applying the softmax function, as softmax$(-\infty) = 0$. In contrast, during NAT model training, none of the true tokens are fed into the decoder, and the model learns to generate all positions from "scratch." Since the model's objective is not a conditional probability like in AT models, the decoder does not use "masking" to prevent information leakage. This approach also

allows the model to be bidirectional, as each position can attend to both preceding and succeeding positions.

2. **Token-Level Loss vs. Sequence-Level Loss:** AT model training uses cross-entropy loss on each token, meaning that the ”reward” signal is applied to each correctly predicted token. On the other hand, NAT models apply the connectionist temporal classification (CTC) loss, as detailed in the manuscript. CTC’s reward signal is applied to entire ”paths,” meaning that partially predicted tokens do not receive a reward. Only entirely correct paths receive a positive reward, while any incorrect path receives a negative reward. This loss mechanism better enhances global sequence coherence.

3. **Decoding Dynamics:** Since NAT models generate probabilities independently at each position, the resulting probability distribution is static. However, in AT models, each token’s probability is conditional, making it dynamic. At each position of an AT model, the selection of a particular token affects the distribution of all subsequent positions. For instance, selecting ”good” at the first position might increase the probability of ”bye” in the second position, while selecting ”see” might increase the probability of ”you” in the second position. This dependency prevents the model from having accurate global control over the generational output, as any modification to a previous token requires re-decoding all subsequent tokens due to the shifted distribution. In contrast, because the NAR model’s probability is static, any position can be modified without such concerns, which is the motivation behind the Precise Mass Control (PMC) unit.

With respect to the NAT model configuration, beyond what is included in Supplementary Information section 2.3:
PrimeNovo utilizes an encoder-decoder transformer network configuration. More precisely, both the encoder and decoder components consist of 9 multi-head attention layers. Each attention layer within the network is composed of 8 heads. The hidden embedding dimension for our model is set to 400.

For the training process, we employed a Cosine learning rate scheduler in conjunction with the AdamW optimizer. To mitigate the risk of excessive overfitting, we set the dropout rate at 0.18.

During inference, we use the PMC with precision ϵ of 0.1Da. This precision constraint ensures that the decoded peptide always falls within a range of 0.1 Da from the precursor mass indicated in the spectrum.

We have further added the following information to the same section:
Our model utilizes a vocabulary of size 28, including 27 amino acid-related tokens: ”G”, ”A”, ”S”, ”P”, ”V”, ”T”, ”C+57.021”, ”L”, ”I”, ”N”, ”D”, ”Q”, ”K”, ”E”, ”M”, ”H”, ”F”, ”R”, ”Y”, ”W”, ”M+15.995”, ”N+0.984”, ”Q+0.984”, ”+42.011”, ”+43.006”, ”-17.027”, and ”+43.006-17.027”, along with one CTC placeholder token ϵ.

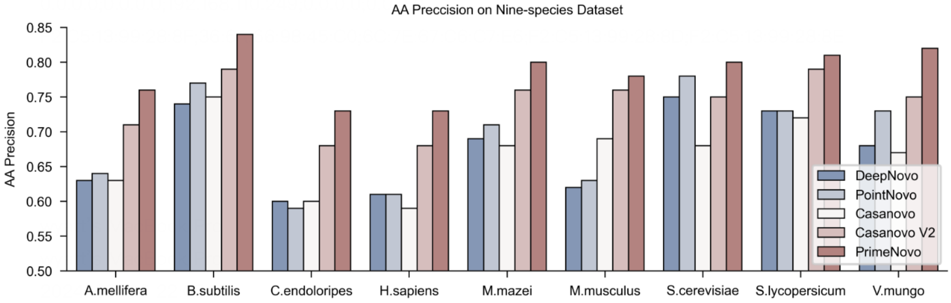
The feedforward layer in the Transformer has a dimension of 1024. We have fixed the maximum generational length to 40 tokens, which will be reduced using CTC post-reduction. Training is conducted over a maximum of 150 epochs, with each epoch covering the entire training dataset. We use the nine-species Bacillus validation set for all training validation, as we believe it can reflect the model’s generalization on out-of-distribution data. Training is stopped when the validation loss converges, which is indicated by minimal changes (+- 0.05) or an increase (overfitting). The last checkpoint before convergence is used for testing. Our PMC unit functions as the post-generational decoding module and does not require additional training or tuning.

All of this information can also be referenced in the ‘config.yaml‘ file in our codebase and the ‘model.py‘ file for more detailed implementation specifics.

2. Performance Evaluation:

2.1 While the reported high recall is commendable, incorporating additional metrics like precision would provide a more comprehensive and comparable evaluation, aligning with common practices in de novo sequencing benchmarking.

Re: We thank the reviewer for this suggestion. We have included amino acid-level (AA) precision for all the tested datasets, which is listed in both the main section and Supplementary Information. For instance, amino acid level precision for nine-species is included in Supplementary Figure 6. For peptide-level recall, since a peptide is considered correct only if all amino acids are correctly predicted, its precision and recall are the same value.



Supplementary Figure 6: The prediction precision at AA level on the nine-species benchmark dataset. The AA precision of PrimeNovo is significantly higher than that of other de novo methods.

This can be further explained through the formulas for Precision and Recall:

- **Precision**:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

- **Recall**

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

Recall and precision are the same when the number of false positives (FP) is equal to the number of false negatives (FN). In mathematical terms, this condition can be expressed as:

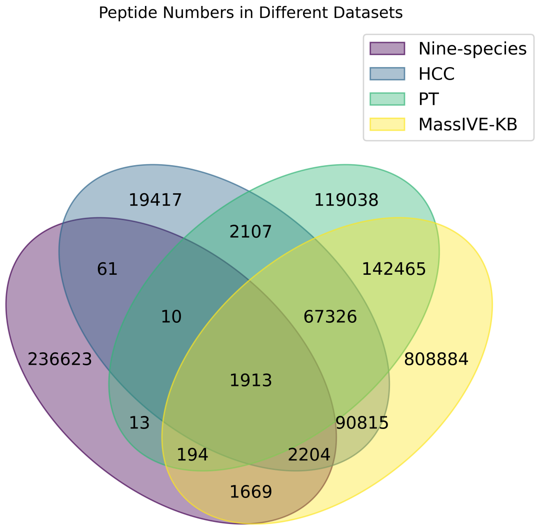
If FP = FN, then:

$$\text{Precision} = \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

In the case of peptide-level evaluation, both False Negatives (FN) and False Positives (FP) refer to incorrectly predicted sequences, as there is no concept of partial correctness or ”fake” correctness. A peptide is considered correct only if all its amino acids are accurately predicted; otherwise, it is deemed incorrect. As a result, peptide recall is exactly the same as peptide precision. Following the convention of all previous work, we refer to this metric as peptide recall, but one can refer to it as peptide precision as well.

2.2 The visualization in Figure 3g could be enhanced with a Venn diagram to offer clearer insights into the generalization capabilities across different datasets.

Re: We have made the diagram according to reviewer’s suggestion and incorporated it into the supplementary Information, in Supplementary Figure 15.



Supplementary Figure 15: Venn diagram of peptide numbers in different datasets (no modification).

3. Experimental Details and Reproducibility.

3.1 The typo in line 317 should be corrected.

Re: We thank the reviewer for their careful reading and for pointing this out. We have corrected the typo.

3.2 Detailed information about the database search software, configuration settings, and relevant references should be provided for the quality control process. A clear description of any pre-processing steps applied to the MassIVE-KB dataset is essential.

The initial construction of the Massive-KB spectral library [1] utilized 31TB of human HCD data from 227 public datasets. Tandem mass spectra were searched using MSGF+ [18]. The allowed variable modifications included methionine oxidation (M+15.995), N-terminal acetylation (+42.011), N-terminal carbamylation (+43.006), pyro-glutamic acid formation on glutamine (Q-17.027), and deamidation on asparagine (N+0.984) and glutamine (Q+0.984). Carbamidomethylation was treated as a fixed modification on cysteine (C+57.021). Each individual search was filtered at a 1% PSM-level FDR. Subsequently, a subset library was selected from the truncated results, retaining up to 100 PSMs per precursor, each with a uniformly 0% q-value in its original search. To obtain the aforementioned dataset, we downloaded the raw files (in mzML format) and the filtered identification results from the "All Candidate library spectra" section of the MassIVE Knowledge Base spectral library v1 were obtained by downloading from the MassIVE repository. Then MSConvert (version 3) were used to convert these raw files to mgf files. Next, we incorporated the peptide sequences, charge states, and precursor ion masses from the identification results into the mgf files.

3.3 Uploading the trained model and all associated results would significantly enhance reproducibility. * The provided code should be thoroughly checked for errors and dependencies to ensure its functionality.

No trained models for PrimeNovo were provided. The provided code cannot be executed without the correct model file. Additionally, there are bugs in the code, such as 'ModuleNotFoundError: No module named 'PrimeNovo.components.encoders"'. Please ensure all code is functional and error-free.

Re: We thank the reviewer for the suggestion and please refer to the general response above for this concern.

Reviewer 2:

The manuscript described a new AI based de novo peptide sequencing tool Pi-PrimeNovo. The main novelty compared to other similar tools is the use of a non-autoregressive model. In addition, it uses a precise mass control to ensure the generated peptide matches the precursor mass of the spectrum. There are other attempt to match the precursor match in AI-based de novo sequencing, but the method used here is new.

The model has been extensively benchmarked on large mass spectrometry datasets. The performance of PrimeNovo exceeds the state of the art significantly in terms of both de novo sequencing accuracy and speed. It also demonstrates excellent accuracy in detecting PTMs.

Re: We thank the reviewer for acknowledging our contributions and emphasizing the novelty and outstanding performance of our work. We appreciate the time you have taken to review our manuscript. We have provided a point-by-point response to address your concerns and comments.

1. The main contribution of the paper is the software. The availability of the software will be a great addition to the proteomics field. In terms of the software availability, I do not see the manuscript discuss the availability of the software. The code and software submission form claimed the software is MIT license. The code is also attached in the reviewer's package. However, I do not see the model's weights file. The model cannot run without the weights file. The training of the weights is a critical part of a neural network model. The re-usability of their results is severely limited if the weights were not to be published.

Re: Thank you for the suggestions and we have addressed this concern and please refer to the general response above.

2. p3, line 68-70: Should the ”all spectra” be ”all assigned spectra”? If a spectrum is not identified with database search, it is hard to know ground-truth sequence for the spectrum and is often excluded from de novo sequencing benchmark.

Re: Thank you for your careful reading and for pointing this out. Yes, spectra must be annotated with database-searched peptide results; otherwise, they cannot be used for training or evaluating the model. We have corrected this in the manuscript.

3. Since speed is an important part of the contribution, provide the de novo sequencing speed in absolute terms (in addition to the ratio to Casanovo).

Re: Indeed, absolute speed is an important consideration. While our main manuscript uses relative comparison (i.e., percentage improvement) between models, Figure 2.d in main manuscript reports the speed in absolute term (spectra/s). Specifically, it shows how many spectra each model can decode per second. This absolute value is calculated under fair conditions, using the same GPU (an Nvidia A100-80GB) for testing. Data loading and model initialization times are excluded to avoid distractions in measuring processing time. We have made the updated Figure 2.d in main manuscript after experimenting the optimal configurations for the inference, as suggested by reviwer 3. As seen in Figure 2.d, our model can decode up to 714 spectra per second on a single A100-80GB GPU.

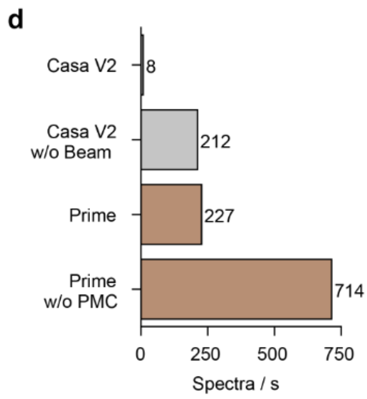


Figure 2.d in Main Manuscript: Updated speed comparison.

4. Figure 3c and 3g Provide a little more explanation about the figure.

Re: Figure 3.c shows the precision-coverage curve for two testing datasets, IgG1-Human-HC and HCC, comparing PrimeNovo and Casanovo. For each model, predictions are ranked based on their confidence scores, with higher scores indicating greater certainty about the correctness of the predictions. We calculate amino acid-level precision based on the percentage of predictions covered within each confidence ranking range. For example, the 0-20% range includes the lowest 20% of predictions based on confidence scores, and we compute the average amino acid precision for these predictions. Conversely, the 80-100% range represents the top 20% of predictions, and we compare the amino acid precision for these high-confidence predictions between models. As shown in Figure 3.c, the top 20% of highly confident predictions are nearly all correct for both models, demonstrating that confidence scores are strong indicators of prediction reliability. PrimeNovo exhibits higher prediction accuracy at lower confidence levels, indicating that its predictions are more reliable even when the model is less confident, compared to Casanovo.

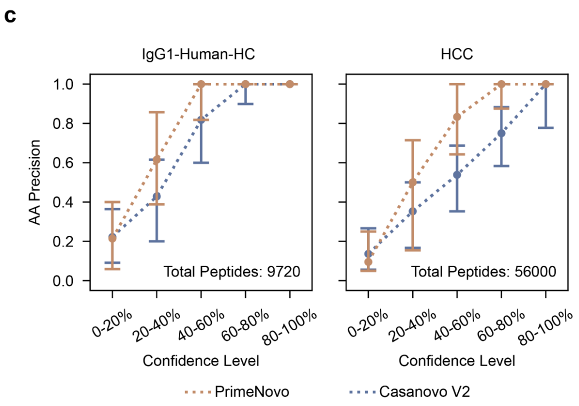


Figure 3.c in Main Manuscript: **c. Amino Acid-Level Precision**: The chart depicts the amino acid-level precision for PrimeNovo and Casanovo V2 on the IgG1-Human-HC and HCC datasets. The x-axis shows the coverage rate of predicted peptides based on each model's confidence score. For instance, 20%-40% indicate the 20%-40% least confident predictions based on confidence scores. AA precision is then calculated within each coverage range.

Figure 3.g illustrates how models trained on different datasets generalize to other datasets, highlighting the generalizability and quality of each training data. Specifically, the left side of Figure 3.g shows models trained on different training data–PT, HCC, MassiveKB, and nine-species datasets, while the right side displays the testing data. For example, the model trained on PT data (green color) performs best on its own PT test set, as indicated by the strongest green line pointing towards its own testing dataset. In contrast, it performs poorly on the nine-species test set, as shown by the thin green line pointing towards it. The number 56% at the green stem represents the average peptide accuracy across all four testing datasets, serving as a measure of the training data's transferability and generalizability. As depicted in

Figure 3.g, the model trained on the MassiveKB dataset demonstrates the best generalizability, achieving the highest accuracy on all other test datasets. This underscores the importance of selecting high-quality, generalizable training datasets to ensure that the model's capabilities can be effectively transferred to different data distributions.

We have made more detailed description in the caption in our revised manuscript for these 2 figures, as seen in the captions of Figure 3.c and Figure 3.g.

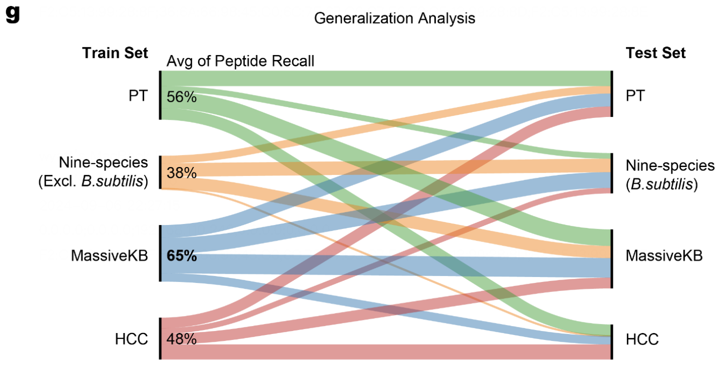


Figure 3.g in Main Manuscript: **g.** This diagram demonstrates the model's generalization capability when trained exclusively with each training dataset. The left-hand side indicates each one of the four training data PrimeNovo is trained on. The thickness of each line indicates the performance on each of the four testing sets on the right-hand side, with a thicker line being better performance. The numbers on the stem indicate the averaged peptide recall over all four testing sets, highlighting the distributional transferability of each training data. The model trained on MassIVE-KB exhibited the highest average peptide recall, 65% (bolded)

5. p10, line 315: intermediate fragment ions → internal fragment ions

Re: We have fixed this, thanks again for pointing this out.

6. p10. line 351: rigorous quality control process T U D DS

What is T/U/D/DS?

Re: The T/U/D/DS is our proposed quality control process. It mainly involves the following steps: first, we identify sequences present in the **T**arget database. Then, we filter out results that are: (1) **U**nmatched with the precursor mass (mass error larger than 0.1 Da); (2) found within the **D**ecoy database; (3) identified in **D**atabase **S**earch results. Both the target and decoy databases were provided in the original study [14].

7. Supplementary Information, section 2.1 CTC loss calculation with dynamic programming.

Will the use of this dynamic programming based loss function affect the back-propagation algorithm for training the neural network? Please provide a brief explanation.

Re: Thank you for your insightful question regarding the loss calculation. The use of dynamic programming for calculating CTC loss does not affect the backpropagation algorithm. Dynamic programming is employed to facilitate the identification of all valid CTC paths, but it does not alter the nature of loss calculation.

To illustrate this, consider an example where we optimize a network with the target label ”AGC” across 4 generational positions. Naively calculating CTC loss requires enumerating all possible paths that can be reduced to ”AGC,” such as ”AGGC,” ”AAGC,” ”ϵAGC,” and many others. For 20 tokens and 1 ϵ placeholder token used by CTC, across 4 positions, this results in 21^4 possibilities, leading to exponential growth in the number of paths as the length increases.

Dynamic programming is introduced to address this complexity. For example, the first position can only be ϵ or A, as only these can contribute to the target ”AGC.” Once the first position is determined, the subsequent positions can be inferred based on the first position using CTC recursion rules, rather than exhaustively enumerating all possibilities for each amino acid. This reduces the complexity of identifying all correct paths but does not change the loss calculation itself.

The loss is still computed as the sum of probabilities for all correct paths. Dynamic programming simply provides an efficient method to find these paths, thereby saving computational time without altering the nature of the loss or the backpropagation process.

We have added a concise paragraph summarizing the above clarification in the Supplementary Information Section 2.1:
It’s worth noting that dynamic programming is employed to efficiently compute the CTC loss by identifying all valid paths without affecting the backpropagation algorithm. Instead of naively enumerating all possible paths that can be reduced to a target label (which grows exponentially), dynamic programming simplifies the process by applying CTC recursion rules, reducing complexity. This method finds all valid paths more efficiently but does not alter the loss calculation, which remains the sum of probabilities for the correct paths.

Reviewer 3:

In this paper, the authors present a new transformer architecture for peptide De novo sequencing from MS/MS data. The new architecture together with CTC loss functon(which is often used in the voice recognition task) seems applied to the peptide De novo sequencing task successfully. The non-autoregressive model architecture gains significant performance improvement over the autoregressive models like Casanovo. The authors tested their model on some previously published benchmark datasets together with some other datasets for special applications. Different metrics are used for all the dataset. The results seem convincing. Generally I recommend publication for this paper. However, there are several major issues requiring attention before the publication

Re: We appreciate the reviewer's confirmation of our work and the performance we have achieved. As pointed out by the reviewer, while CTC-based models have been widely used in speech recognition and generation, we are among the first to successfully apply them to biological research. We are grateful for the recommendation for publication and have included a detailed point-by-point response to address your concerns and questions.

1. Although source code is provided for the review, the models are not provided which makes it hard to reproduce their results. If models are not provided, at least some description should be included for the training process and hyper parameter settings.

Re: We have addressed the concern of model availability in the general response above.

Regarding model parameters, we have expanded the "Model Configuration" in the Supplementary Information Section 2.3 as follows:
PrimeNovo utilizes an encoder-decoder transformer network configuration. More precisely, both the encoder and decoder components consist of 9 multi-head attention layers. Each attention layer within the network is composed of 8 heads. The hidden embedding dimension for our model is set to 400.

For the training process, we employ a Cosine learning rate scheduler in conjunction with the AdamW optimizer. To mitigate the risk of excessive overfitting, we set the dropout rate at 0.18.

During inference, we use the PMC with precision ϵ of 0.1Da. This precision constraint ensures that the decoded peptide always falls within a range of 0.1 Da from the precursor mass indicated in the spectrum.

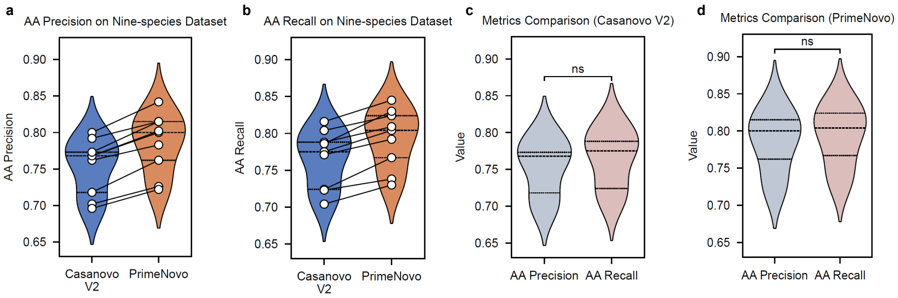
Our model utilizes a vocabulary of size 28, including 27 amino acid-related tokens: ”G”, ”A”, ”S”, ”P”, ”V”, ”T”, ”C+57.021”, ”L”, ”I”, ”N”, ”D”, ”Q”, ”K”, ”E”, ”M”, ”H”, ”F”, ”R”, ”Y”, ”W”, ”M+15.995”, ”N+0.984”, ”Q+0.984”, ”+42.011”, ”+43.006”, ”-17.027”, and ”+43.006-17.027”, along with one CTC placeholder token ϵ.

The feedforward layer in the Transformer has a dimension of 1024. We have fixed the maximum generational length to 40 tokens, which will be reduced using CTC post-reduction. Training is conducted over a maximum of 150 epochs, with each epoch covering the entire training dataset. We use the nine-species Bacillus validation set for all training validation, as we believe it can reflect the model's generalization on out-of-distribution data. Training is stopped when the validation loss converges, which is indicated by minimal changes (+- 0.05) or an increase (overfitting). The last checkpoint before convergence is used for testing. Our PMC unit functions as the post-generational decoding module and does not require additional training or tuning.

Detailed information, including all model hyperparameters and optimal settings, is also available in the `config.yaml` file in our provided code base.

2. Peptide recall is used for most data set evaluations. However, amino acid recall on all identified psms is a more robust metric to avoid the overfitting problem for the deep neural network models. It should be also included

Re: We have included Amino Acid Precision for all tested datasets in our original Supplementary Information. We did not initially include Amino Acid Recall for all datasets because we observed that these two metrics are very close to each other for all models tested, including ours. As the reviewer pointed out, we have now included Amino Acid Recall for comparison on our main benchmark nine-species test set for demonstration (Supplementary Figure 3). Additionally, we have incorporated Supplementary Figure 3.b into Figure 2.c of the main manuscript for a more straightforward comparison at the AA level accuracy. And we have added the following content to the revised Supplementary Information to further illustrate the relations between these two metrics:

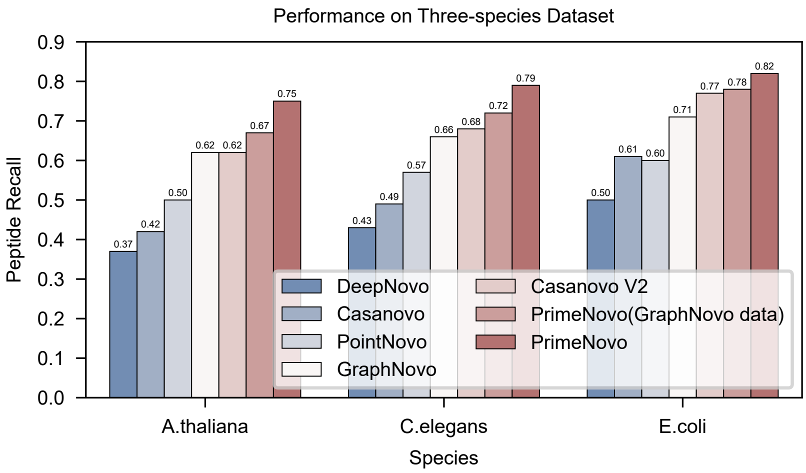


Supplementary Figure 3: **Relationship between AA recall and AA precision.** **a**. Comparison of AA precision across 9 different test species between PrimeNovo and Casanovo V2. **b**. Comparison of AA recall across the same 9 test species between PrimeNovo and Casanovo V2. **c**. The difference in distribution between AA recall and AA precision for Casanovo V2 across the nine-species dataset (ns indicates `not significant` (p-value set to 0.05)) according to the Mann-Whitney U test). **d**. The difference in distribution between AA recall and AA precision for PrimeNovo across the nine-species dataset (ns indicates `not significant` according to the Mann-Whitney U test).

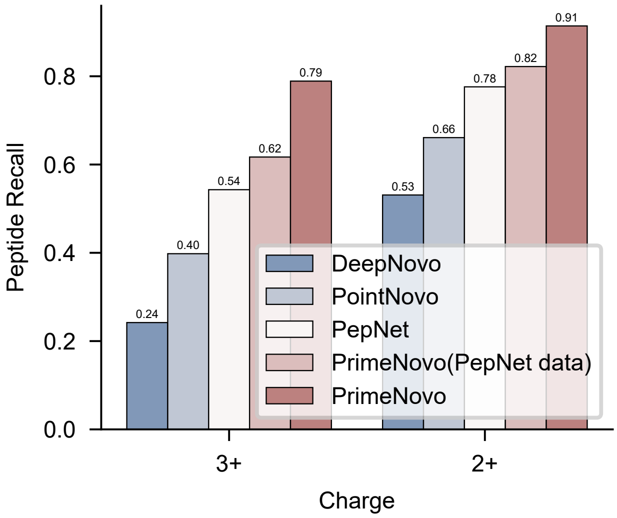
As shown in Supplementary Figure 3.a-b, PrimeNovo consistently outperforms Casanovo V2 in both AA precision and AA recall across all species. For both models, we observe that AA recall and AA precision are very similar, with AA recall being, on average, 0.5%–1% higher than AA precision. We further investigated the difference between AA precision and AA recall using statistical testing. As shown in Supplementary Figure 3.c-d, the Mann-Whitney U test indicates no significant difference (p-value $<$ 0.05) between recall and precision across all species, with the mean recall value slightly higher (less than 1%) than precision. This lack of significance is consistent for both PrimeNovo and Casanovo V2, with PrimeNovo showing even smaller differences between recall and precision (Figure Supplementary Figure 3.d) and higher confidence level (lower p-value) in rejecting the non-difference using Mann-Whitney U test. In conclusion, these two metrics do not show substantial differences in representing Amino Acid level prediction performance and can be used interchangeably. Thus, our reported AA precision across all datasets serves as a robust evaluation of AA-level accuracy.

3. For the GraphNovo and PepNet, to make it a fair comparison, better use the same training data for different models.

Re: Thank you to the reviewer for pointing this out. Indeed, using the same training data is necessary for a fair comparison. We have re-trained our model using the training data released by GraphNovo and the training data provided by PepNet's authors. The results are shown in Rebuttal Figure 3 and Rebuttal Figure 4. As shown, when trained on the same dataset, PrimeNovo demonstrates superior accuracy compared to GraphNovo, with peptide recall improving from 62% to 67%, 66% to 72%, and 71% to 78% in *A. thaliana*, *C. elegans*, and *E. coli*, respectively (Rebuttal Figure 3). Additionally, our model, trained on the GraphNovo dataset, outperforms Casanovo V2, which was trained on the much larger and more diverse MassiveKB dataset. When trained on the PepNet dataset, PrimeNovo further surpasses PepNet, improving PepNet's 54% peptide call to 62% in peptides with 2+ charges and 78% to 82% in peptides with 3+ charges. These results provide strong evidence of PrimeNovo's superior performance and robust design, which excels in all fair comparisons.



Rebuttal Figure 3: Peformance comparison on 3species testing dataset used by GraphNovo. PrimeNovo (GraphNovo data) is PrimeNovo retrained with GraphNovo training data.



Rebuttal Figure 4: Peformance comparison on test dataset used by PepNet. PrimeNovo (PepNet data) is the PrimeNovo retrained using PepNet training dataset.

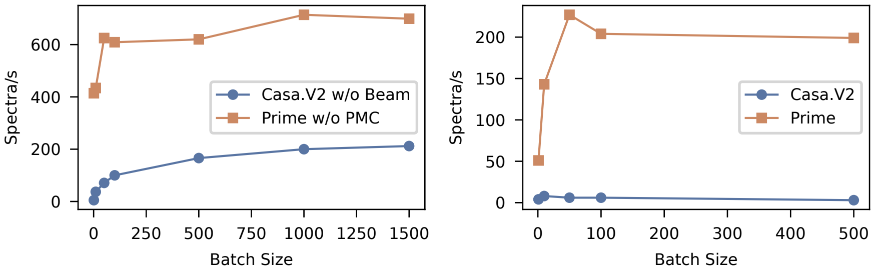
4. For speed comparison with Casanovo V2, the Casanovo speed seems quite inconsistent with our own experience. The authors better contact Casanovo authors to make sure their parameter settings are optimised for Casanovo.

Re: For the speed testing, we have noticed that not only the model's configuration, but also factors such as the testing machine, current CPU offload, memory space, and other conditions can affect the final results. We have made every effort to ensure that the speed testing was conducted under fair conditions. This includes:

1. Using the exact same GPU and GPU node for testing both models, so all tests were conducted under the same computational environment.
2. Eliminating all GPU warmups, model pre-loading and data processing times from the model's evaluation, as these are largely affected by memory and CPU usage, as well as threading allocations. We only measured the time from when the model starts inference on the GPU to the time when results are returned.

3. Repeating the experiments multiple times to account for variance due to random factors.

Regarding the model's configuration, we adhered to the provided configuration file from Casanovo, and our results in performance align well with those reported in the original paper. However, as the reviewer noted a potential discrepancy between our reported speed and their replication, we conducted further experiments to investigate this. We identified that a significant factor affecting evaluation speed was the batch size. The results reported in our paper used a fixed batch size for both models, which led to an underestimation of the speed for both PrimeNovo and Casanovo. We conducted detailed experiments and observed that inference batch size is the most critical factor influencing speed. Our ablation study on the impact of batch size on inference speed is presented in Supplementary Figure 2. We therefore added the following content in our Supplementary Information to disclose our findings :



Supplementary Figure 2: Relationship between inference speed (spectra/s) and inference batch size. The left-hand side shows results when no post-decoding algorithms (PMC and beam search) are used for either model, while the right-hand side shows results with PMC and beam search applied.

We observed that a larger batch size does not always result in faster inference. We discuss the cases where post-modification processes—PMC for PrimeNovo and beam search for Casanovo v2—are enabled, compared to when these processes are disabled.

- When both post-modification processes were included, we found that the optimal inference speed was achieved with a batch size between 10 and 50 for both models (Supplementary Figure 2 right hand side). Increasing or decreasing the batch size beyond this range resulted in reduced speed. Specifically, Casanovo with beam search reached its peak performance at a batch size of 10, processing 8 spectra per second. PrimeNovo with PMC achieved its best performance at a batch size of 50, processing 227 spectra per second—28 times faster than Casanovo V2. The small optimal batch size is due to the heavy reliance of both post-decoding algorithms on CPU processing after our investigation, which involves many non-parallelizable components and conditional branches (e.g., if statements). Larger batch sizes lead to CPU offloading inefficiencies beyond the optimal threshold.
- When both post-decoding algorithms were disabled, we observed a continuous increase in speed with larger batch sizes (Supplementary Figure 2 left hand side). Specifically, both models reached their speed plateau at a batch size more than 1000. In this case, PrimeNovo without PMC was able to decode up to 714 spectra per second, while Casanovo without beam search reached 212 spectra per second. This is because, without the post-decoding algorithms, both models primarily rely on GPU processing, which efficiently handles large amounts of parallel work. Larger batch sizes improve GPU utilization.

We have updated the optimal speed for both models in the main manuscript's Figure 2.d based on these findings. Additionally, we tested other factors such as the number of spectra peaks and GPU card numbers and found minimal impact on speed, indicating that batch size is likely the primary factor affecting performance. And we believe our updated results align with what the reviewer has obtained.

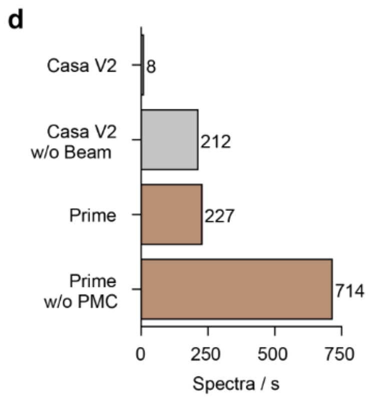
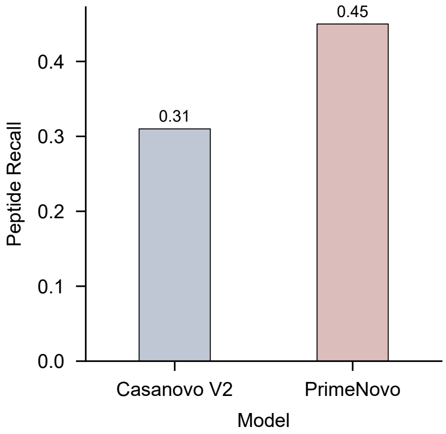


Figure 2.d in Main Manuscript: Updated speed comparison.

5. All the data in this paper are acquired through the Thermo orbitrap instrument. It needs to include the data from other instruments, e.g.,timsTOF Pro, and with some comparison and discussion for the performance.

Re: Thanks for the reviewer's suggestion. We have added following results into our Supplementary Information:
The mass spectrometry data generated by different types of experimental instruments can exhibit significant differences. Models trained on mass spectrometry data sourced from Thermo orbitrap instrument may not necessarily perform effectively on other types of instruments. To test our method's robustness against spectrum data from different instruments, we have chosen a public DDA-PASEF dataset [19] generated from timsTOF Pro instrument to evaluate our model's performance. The raw files (in `.d` format) and MSFragger identification results were obtained by downloading from the PRIDE repository by PXD041421. Mzml files generated by MSFragger were converted to mgf files and added identified information. This dataset contains 3667330 PSMs with 83272 peptides.

We randomly sampled 5% of the entire dataset for evaluation, with the results shown in Supplementary Figure 14. As observed, PrimeNovo demonstrates the ability to effectively decode spectra generated by the timsTOF Pro instrument. Specifically, it achieved a 45% peptide recall in a zero-shot setting, where no fine-tuning was performed on this data distribution. In comparison, Casanovo V2 achieved a peptide recall of 31%, significantly lower than our approach. This 14% improvement over Casanovo V2 is consistent with the performance advantage observed in other datasets (ranging from 10% to 30%).



Supplementary Figure 14: Performance of PrimeNovo and Casanovo V2 on a timsTOF Pro based dataset.

References

[1] Wang, M. *et al.* Assembling the Community-Scale Discoverable Human Proteome. *Cell Systems* **7**, 412–421.e5 (2018).

[2] Tran, N. H., Zhang, X., Xin, L., Shan, B. & Li, M. De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences* **114**, 8247–8252 (2017).

[3] Zolg, D. P. *et al.* Building ProteomeTools based on a complete synthetic human proteome. *Nature Methods* **14** (2017).

[4] Vizcaíno, J. A. *et al.* 2016 update of the PRIDE database and its related tools. *Nucleic acids research* **44**, D447–56 (2016).

[5] Paul Zolg, D. *et al.* Proteometools: Systematic characterization of 21 post-translational protein modifications by liquid chromatography tandem mass spectrometry (lc-ms/ms) using synthetic peptides. *Molecular and Cellular Proteomics* **17**, 1850–1863 (2018).

[6] Müller, J. B. *et al.* The proteome landscape of the kingdoms of life. *Nature* **582**, 592–596 (2020).

[7] Liu, K., Ye, Y., Li, S. & Tang, H. Accurate de novo peptide sequencing using fully convolutional neural networks. *Nature Communications* **14** (2023).

[8] Jiang, Y. *et al.* Proteomics identifies new therapeutic targets of early-stage hepatocellular carcinoma. *Nature* **567**, 257–261 (2019).

[9] Ma, J. *et al.* iProX: an integrated proteome resource. *Nucleic acids research* **47**, D1211–D1217 (2019).

[10] Xu, J. Y. *et al.* Integrative Proteomic Characterization of Human Lung Adenocarcinoma. *Cell* **182**, 245–261.e17 (2020).

[11] Tran, N. H. *et al.* Complete de Novo Assembly of Monoclonal Antibody Sequences. *Scientific Reports* **6**, 1–10 (2016).

[12] Mao, Z., Zhang, R., Xin, L. & Li, M. Mitigating the missing-fragmentation problem in de novo peptide sequencing with a two-stage graph-based deep learning model. *Nature Machine Intelligence* **5**, 1250–1260 (2023).

[13] Yilmaz, M. *et al.* Sequence-to-sequence translation from mass spectra to peptides with a transformer model. *Nature communications* **15**, 6427 (2024).

[14] Patnode, M. L. *et al.* Interspecies competition impacts targeted manipulation of human gut bacteria by fiber-derived glycans. *Cell* **179**, 59–73 (2019).

[15] Qiao, R. *et al.* Computationally instrument-resolution-independent de novo peptide sequencing for high-resolution devices. *Nature Machine Intelligence* **3**, 420–425 (2021).

[16] McIlwain, S. *et al.* Crux: Rapid Open Source Protein Tandem Mass Spectrometry Analysis. *Journal of Proteome Research* **13**, 4488–4491 (2014).

[17] Yilmaz, M., Fondrie, W., Bittremieux, W., Oh, S. & Noble, W. S. De novo mass spectrometry peptide sequencing with a transformer model. In *Proceedings of the 39th International Conference on Machine Learning*, vol. 162, 25514–25522 (2022).

[18] Kim, S. & Pevzner, P. A. MS-GF+ makes progress towards a universal database search tool for proteomics. *Nature Communications* **5**, 1–10 (2014).

[19] Wang, H. *et al.* MultiPro: DDA-PASEF and diaPASEF acquired cell line proteomic datasets with deliberate batch effects. *Scientific Data* **10**, 858 (2023).

Response to Reviewers

Manuscript ID: NCOMMS-24-37565A

Manuscript title: π-PrimeNovo: An Accurate and Efficient Non-Autoregressive Deep Learning Model for De Novo Peptide Sequencing

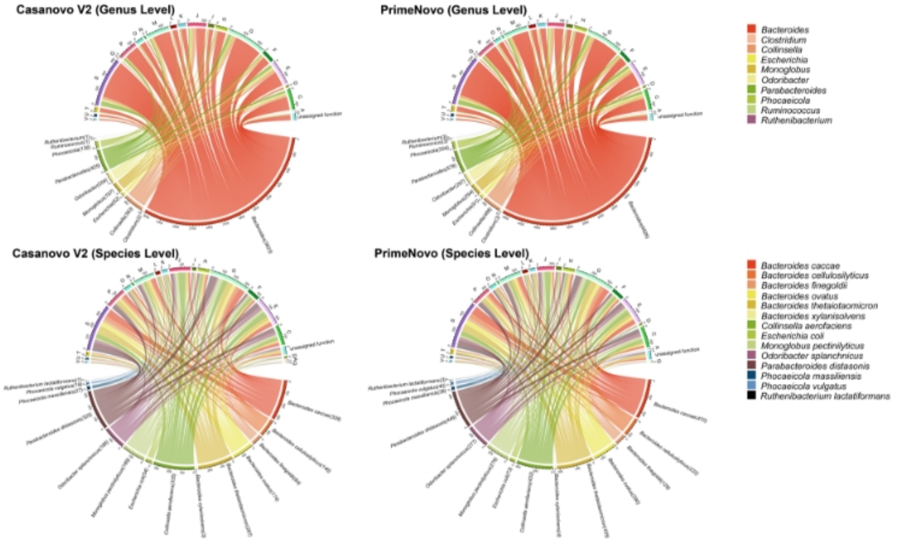
We sincerely thank reviewers reading our previous reply and giving prompt reply back. We have further made comments on all the questions reviewers have raisied this time and we include a point by point response, as before, in following.

- Blue: Reviewer's comments.
- Black: Our response.
- Orange: Reference to the (updated) manuscript.

Reviewer 1:

1.Line 364: The manuscript states,"As depicted in Fig. 5f, proteins identified by PrimeNovo and Casanovo were correctly assigned to 10 genera, 14 species, and 20 COG (Clusters of Orthologous Groups of proteins) categories." However, there is no Fig.5f in the manuscript. Please clarify the correct figure reference.

Re: Thank you for the careful reading and for catching this error. During the initial manuscript preparation, we intended to move the original Fig.5f to the Supplementary Information section but inadvertently omitted it at the end. This has now been corrected with the proper reference to the newly added Supplementary Figure 23, which was the missed "Fig.5f".



Supplementary Figure 23: This figure presents the taxonomic and functional annotation results at the protein level. PrimeNovo notably enhances both taxonomic and functional resolution at the genus and species levels compared to Casanovo V2. The corresponding abbreviations for Clusters of Orthologous Groups of proteins (COG) functions are shown in Supplementary Table 11.

2. Peptide-to-Protein Mapping: The performance of the proposed models was evaluated at the protein level, but the process of mapping peptides to proteins is unclear. Could you provide more details on the methodology used for this mapping? Were third-party tools employed, or was an in-house script developed for this purpose? Additionally, what criteria were used to map peptides to proteins?

Re: Thank you for the question and apologize for any lack of clarity. In our paper, we first aligned the peptide sequences inferred by PrimeNovo with the reference proteome database provided in the original paper and recorded the proteins that fully matched each peptide, generating a peptide-protein mapping table. Then we employed an R script that replicates the protein inference algorithm used by MaxQuant (https://github.com/ningzhibin/protein-Inference). The specific workflow of this script is as follows:

- Use the peptide-protein mapping as input and reverse the mapping to create a protein-peptide mapping.
- Sort the table according to the number of peptides mapped to each protein.
- Begin with the top protein (leading protein) and iterate through the table from the second entry onward, identifying all sub-proteins (proteins with peptide IDs that are subsets of the leading protein). Group these as a protein group and store them in a separate table, then remove them from the original table.
- Repeat the previous step until the table is empty.
- Re-examine the list to determine if the peptides are group-unique.
- Remove protein groups with zero razor proteins.

3. Peptides Not Mapped to Proteins: Among the peptides identified by the proposed models, how many were not mapped to any proteins? A discussion on these unmapped peptides would be beneficial. Were they potentially novel peptides with post-translational modifications (PTMs) or protein variants? Alternatively, could they be peptides with amino acid prediction errors?

Re: Thank you for the valuable comment. In Figure 5, we performed de novo sequencing for the unidentified MS/MS spectra by database search from a metaproteomic dataset [1]. After a strict QC process (U\D\DS which filter out results that are: **U**nmatched with the precursor mass (mass error larger than 0.1 Da); Found within the **D**ecoy database; identified in **D**atabase **S**earch results), we found 4,457,223 peptides were not mapped to any proteins. We believe the primary sources of these unmatched peptides can be attributed to two main factors: (1) **Peptides arising from unknown or unannotated protein variants**: The samples may contain genomic variations, such as mutations, insertions, or deletions, which could produce peptides that do not fully align with known reference protein sequences. Additionally, some peptides may derive from rare subtypes or homologous proteins expressed in the samples but are insufficiently annotated in current databases, further increasing the likelihood of mismatches. (2) **Inaccurately predicted peptides**: Although basic quality control measures were applied to the model predictions—such as filtering out peptides with theoretical masses that deviated from the precursor ion mass by more than 0.1 Da—this approach has limitations. For instance, peptides may possess the same amino acid composition as the true sequence but arranged in a different order, resulting in failed matches with the original peptide sequences.

4. Fig. 5b: The peptides under ”PrimeNovo” in Fig. 5b—do these numbers represent peptides that could not be mapped to any bacteria? If so, an explanation of these peptides would be helpful.

Re: Sorry for our unclear descriptions. In Fig. 5b, after taxonomic annotation with Unipept, we removed the peptides not present in the FASTA database and those without an LCA (lowest common ancestor). For PrimeNovo's identified peptides, 1,047 (649+398) peptides were classified as bacterial (PrimeNovo-B), while 2,069 (1486+583) peptides could not be mapped to any bacterial taxa. For Casanovo V2's identified peptides, 520 (122+398) peptides were classified as bacterial (Casanovo V2-B), while 874 (291+583) peptides could not be mapped to any bacterial taxa.

5. Comparison to Database Searching Tools: Is there any discussion comparing the proposed method to traditional database searching tools? Can the proposed model replace conventional database searching, or does it serve a complementary role?

Re: Thank you for your question. PrimeNovo, as a de novo sequencing tool, demonstrates notable improvements in both accuracy and efficiency. We believe it can effectively complement traditional database searches in routine proteomics analysis, as it captures sequence information that conventional database-based methods might miss. In particular, for applications such as analyzing metaproteomic data from unknown species, identifying rare or novel modifications, or discovering new antigens, constructing a suitable reference proteome database can be highly challenging or even unfeasible. In these cases, PrimeNovo's database-independent approach provides a distinct advantage, making it particularly well-suited for these specialized tasks. However, current de novo sequencing approaches still cannot fully replace conventional database searching due to relatively low accuracy.

6. Line 418: The manuscript reports that the classification accuracies for all PTMs exceeded 95%, except for asymmetric and symmetric dimethylation at Arginine (R) and monomethylation at Arginine (R), which have accuracies of 77%, 77%, and 69%, respectively. Could the authors provide an explanation for why these specific PTMs exhibit lower accuracies compared to others? Furthermore, given that recall rates are comparable to other datasets without special PTMs, does this suggest that fine-tuning on PTM-enriched datasets may not be necessary?

Re: Thank you for this valuable question. Due to the complexity of deep learning models, pinpointing the exact causes of performance disparities can be challenging. Many practical factors, such as data distribution, optimization landscape complexity, and model architecture bias, can contribute to these differences, the investigation of these falls out of our research scope.

For above-mentioned 3 PTMs with low performance, we believe that the identification of methylation based on mass spectrometry itself presents more challenges, as there are interferences from modifications with the same mass, such as the substitution of Aspartic acid (Asp) with Glutamic acid (Glu), Glycine (Gly) with Alanine (Ala), Serine (Ser) with Threonine (Thr), Valine (Val) with Leucine (Leu) or Isoleucine (Ile), and Asparagine (Asn) with Glutamine (Gln). The mass difference between these modifications is exactly equivalent to one methyl group. Therefore, when PrimeNovo analyzed these three datasets, it may mistakenly identify the amino acid substitutions as the presence of methylation modification, ultimately leading to a decrease in overall accuracy.

Regarding the second part of the question: Fine-tuning is essential when a PTM is absent from the MassiveKB data, as the model has not encountered these PTMs during pretraining and thus needs fine-tuning to learn their occurrence patterns. Using a more PTM-enriched and diverse fine-tuning dataset is beneficial, as it would expose the model to a broader range of patterns, helping it capture the true distribution of these PTMs.

7. Fine-Tuning for Phosphorylation Dataset: The manuscript mentions that PrimeNovo was fine-tuned on a training dataset and tested on phosphorylation data from Xu et al., achieving a classification accuracy of 98% and a peptide recall rate of 66%. Why was additional fine-tuning applied specifically for the phosphorylation dataset, given that the model had already been fine-tuned on the 21 PTMs dataset? What would the performance have been on the phosphorylation dataset without this additional fine-tuning? Is fine-tuning always required in practice when applying the model to PTM analysis?

Re: (1) ”why additional fine-tuning”: The 21PTM dataset exhibits relatively low diversity, with around 700,000 PSMs, averaging 20,000–30,000 PSMs per PTM (Supplementary Table 2). The high accuracy observed on this dataset likely results from the similarity between training and testing distributions, as both training and testing data were collected from the same samples. Although the 21PTM dataset demonstrates that PrimeNovo can adapt to various PTMs, this result does not ensure generalization to more diverse, real-world datasets due to its smaller sample size and limited diversity. Therefore, we conducted additional fine-tuning on the phosphorylation dataset from Xu et al. to further validate PrimeNovo's adaptability on data with much higher diversity and complexity.

(2) ”results without fine-tuning”: Without fine-tuning, when we applied the model trained on 21PTM data to Xu et al.'s phosphorylation dataset, PrimeNovo achieved a peptide recall of only around 20%. This lower recall reflects the increased diversity in the Xu dataset, which includes over 25 million PSMs collected from cancer patients, representing a broader range of phosphorylation PTMs. This substantial drop in performance highlights the importance of fine-tuning when adapting the model to datasets with different underlying distributions and greater diversity.

(3) ”is fine-tuning always required”: Yes, for PTM analysis, fine-tuning is necessary for PrimeNovo if the PTM does not appear in MassiveKB, our pretraining dataset, as the model requires exposure to recognize and adapt to these new PTMs. After fine-tuning on a smaller dataset like 21PTM, additional fine-tuning is still highly recommended, even though the model can recognize the PTM tokens, as its generalization ability remains limited. This limitation follows the current deep-learning scaling law [2], where increased data variety and scale continue to enhance model performance. However, fine-tuning may not be required after training on a large PTM dataset, such as Xu et al.’s phosphorylation data, where scaling effects appear to reach saturation. For a similar experiment, please refer to Figure 3(e) in main manuscript, which discusses performance in relation to fine-tuning dataset size.

8. Hyperparameters: How many hyperparameters were involved in the proposed model? A brief discussion on hyperparameter selection and its impact on the model’s performance would be informative.

Re: Thank you for the comments on this! We address both the architecture and training hyperparameters separately below.

Architecture Hyperparameters: As discussed in Supplementary Note 2 : both the encoder and decoder components consist of 9 multi-head attention layers. Each attention layer within the network is composed of 8 heads. The hidden embedding dimension for our model is set to 400. The feedforward layer in the Transformer has a dimension of 1024. We have fixed the maximum generational length to 40 tokens, which will be reduced using CTC post-reduction. In summary, the architecture hyperparameters include the number of heads (8), embedding dimension (400), number of attention layers (9), feedforward dimension (1024), maximum CTC length (40), and dropout rate (0.18). Most values, except for embedding dimension and dropout rate, follow the previous state-of-the-art Casanovo V2 model, which demonstrated optimal settings for autoregressive Transformer architectures in same tasks. We reduced the embedding dimension (from 512 to 400) and increased the dropout rate (from 0.1 to 0.18) based on early experiments that showed constant overfitting during training; these adjustments effectively mitigated model’s memorization issues and improved performance during validation.

Training Hyperparameters: The training setup, as detailed in Supplementary Note 2: Training is conducted over a maximum of 150 epochs, with each epoch covering the sampled 1 million data in the training dataset. Training is stopped when the validation loss converges, which is indicated by minimal changes (+- 0.05) or an increase (overfitting).. We have added additional training hyperparameters as follows: a base learning rate of 0.0004, with a linear warmup over the first 5 epochs. From the 6th epoch onward, a cosine learning rate scheduler with a 0.5 decay factor is applied. The total batch size is 3200, distributed across 8 GPUs. We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and no additional weight decay (set to 0). In summary, the training hyperparameters include maximum epochs (150), learning rate (0.0004), warmup epochs (5), batch size (3200), Adam optimizer parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$), and cosine learning rate decay factor (0.5).

These updated descriptions have been added in Supplementary Note 2.

9. Line 199: The manuscript states that PrimeNovo had 34,747 overlapping PSMs with MaxQuant search results, while Casanovo V2 and Casanovo had 26,591 and 16,814, respectively. It is encouraging to see that PrimeNovo demonstrates more consistency in its predictions compared to traditional database-searching tools. However, it would be useful to know how many peptides were reported by traditional database-search tools but missed by the de novo sequencing methods. Additionally, how many amino acid mismatches were observed when comparing peptides identified by de novo sequencing to those identified by traditional methods? Discussing the potential limitations of the proposed model and de novo sequencing would provide valuable context.

Re: Thank you for your insightful questions. Regarding the peptides missed by de novo sequencing methods: The de novo sequencing algorithms used provide inferred peptide sequences for each spectrum, accompanied by a confidence score. For comparison, we take the results from a search algorithm with rigorous quality control as the gold standard. Out of 58,528 spectra, the numbers of incorrectly predicted peptide sequences by PrimeNovo, Casanovo V2, and Casanovo are 23,781, 31,937, and 41,714, respectively.

On the Question of mismatched amino acids: In comparison with the 58,528 peptides identified by MaxQuant, the numbers of mismatched amino acids in sequences predicted by PrimeNovo, Casanovo V2, and Casanovo are 228,554, 302,352, and 449,805, respectively. Figure 1 illustrates the distribution of these mismatched amino acids. We see that PrimeNovo has a dominant accuracy advantage when compared with previous methods.

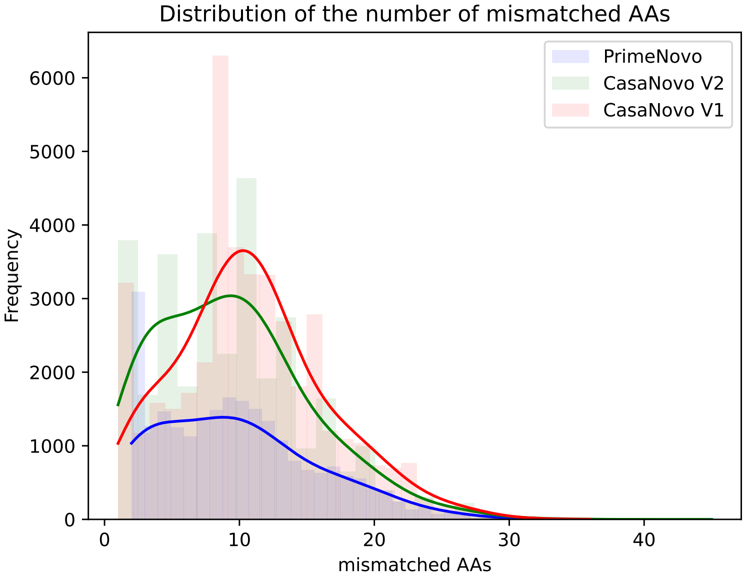


Figure 1: Mismatch amino acids between PrimeNovo and Baseline Models.

Reviewer 2:

Thanks for your kind response! We are glad to hear that your concerns have been addressed!

Reviewer 3:

We appreciate your valuable time and careful reading! Thank you for identifying the error in the formula. We have made corresponding corrections to our manuscript.

References

[1] Patnode, M. L. *et al.* Interspecies competition impacts targeted manipulation of human gut bacteria by fiber-derived glycans. *Cell* **179**, 59–73 (2019).

[2] Kaplan, J. *et al.* Scaling laws for neural language models (2020). URL https://arxiv.org/abs/2001.08361. 2001.08361.