Dear Dr. Hlusko,

We thank you for considering our manuscript as well as the reviewers for taking their time to provide feedback. We are happy to present a revised manuscript for PaleoProPhyler.

In this document we address the comments provided by the reviewers point by point. Our replies to these comments or suggestions will use the following color codes:

- Blue, for the original reviewer's comments.
- Black, for our responses.
- Green, for any text that has been removed, edited or appended.

Our response also includes a word document (.docx) that tracks all changes that were applied to the manuscript, as well as a new version of the manuscript, as a pdf file deposited on Biorxiv, with all the changes implemented. Changes have also been made to the code of the described workflows in order to make its modules more easily executable in different computer environments. Furthermore both the supplementary material and the online tutorial of the workflows have been updated with the goal of providing more details as well as alternative ways of installing and running the workflows.

We hope that the additions and edits are satisfactory and look forward to any additional comments and feedback.

Sincerely,
Ioannis Patramanis on behalf of all the authors.

Anonymous Reviewer 1

Patramanis et al. describe PaleoProPhyler, a pipeline to download, build, and analyze protein sequence databases for phylogenetics including with paleoproteomic sequences. This is an interesting workflow and will help standardize phylogenetics in paleoproteomics.

We thank the reviewer for their interest in our work. One of our main aims is indeed to standardize the phylogenetics workflow in palaeoproteomics.

"files into amino acid seuqences" should be "files into amino acid sequences"

The spelling mistake has been corrected.

Description of the Pipeline: Please add detail/summary of each module here in the main manuscript. The supplementary has nice detail of each module, but it is very lacking here in the main manuscript.

We thank the reviewer for their suggestion, which was made by other reviewers as well. We have thus decided to move some of the description of the workflows from the supplementary into the main text. The following text has been added into the main manuscript:

Module 1 is designed to provide the user with a baseline (curated) reference dataset as well as the resources required to perform the *in silico* translation of proteins from mapped whole genomes. The input of module 1 is a user-provided list of proteins and a list of organisms. The user also has the option of choosing a particular reference build. Utilizing the Ensembl API [1], the module will return 3 different resources for each requested protein and for each requested organism. These are: a) the reference protein sequence of that organism in FASTA format [2], b) the location (position and strand) of the gene that corresponds to that protein and c) the start and end of each exon and intron of that gene / isoform. The downloaded FASTA sequences are available individually but are also assembled into species- and protein-specific datasets. They can be immediately used as a reference dataset for either downstream phylogenetic analyses or as an input database for mass spectrometry software, like MaxQuant [3], Pfind [4], PEAKS [5] and others [6, 7, 8, 9].

The gene location information and the exon / intron tables can be utilized automatically by Module 2. For each requested protein, the module will select the Ensembl canonical isoform by default. Should the user desire a specific isoform or all protein coding isoforms of a protein, they have the ability to specify that as an option in the provided protein list.

Module 2 is designed to utilize the resources generated by Module 1 and to extract, splice and translate genes from whole genome data, into the proteins of interest. This module can handle some of the most commonly used genomic data file formats, including the BAM [10], CRAM [11] and VCF [12] formats. The easiest way to run Module 2 is to first run Module 1 for a set of proteins and a selected organism. This will generate all the necessary files and resources required for the protein translation. This selected organism will be then used as a reference for the translation process. All genomic data to be translated in a single run must be mapped onto that same reference organism. The user can then run Module 2 simply by providing the organism's name (and optionally a reference version), as well as a list of the samples to be translated. The user can also translate samples from a VCF file, but they will need to provide a reference genome in FASTA format, to complement the variation-only information of the VCF file. The translated protein sequences are available individually but are also assembled into individual- and protein-specific datasets.

Module 3 is designed to perform a phylogenetic analysis, with some modifications needed when working with palaeo-proteomic data. The input of this module is a FASTA file, containing all of the protein sequences from both the reference dataset and the ancient sample(s) to be analyzed. The dataset is automatically split into protein specific sub-datasets, each of which will be aligned and checked for SAPs. The alignment is a two step process which includes first isolating and aligning the modern/reference dataset and then aligning the ancient samples onto the modern ones using Mafft [14]. Isobaric amino acids that cannot be distinguished from each other by some mass spectrometers are corrected to ensure the downstream phylogenetic analysis can proceed without issues. Specifically, any time an Isoleucine (I) or a Leucine (L) is identified in the alignment, all of the modern sequences are checked for that position. If all of them share one of the 2 amino acids, then the ancient samples are also switched to that amino acid. If both I and L appear on some present-day samples, both present-day and ancient samples are switched to an L. The user also has the option to provide an additional file named 'MASKED'. Using this optional file, the user can mask a present-day sample such that it has the same missing sites as an ancient sample. Finally a small report is generated for each ancient sample in the dataset, and a maximum likelihood phylogenetic tree is generated for each protein sub-dataset through PhyML [15]. All protein alignments are then also merged together into a concatenated dataset. The concatenated dataset is used to generate a maximum-likelihood species tree [16] through PhyML and a Bayesian species tree [17, 18] through MrBayes [19] or RevBayes [20]. The tree generation is parallelized using Mpirun [21].

Supplementary Choosing and preparing the list of proteins: \Reference_Protein_List.txt is not present in the github repository

Our apologies for missing this file upload. The file has been updated, uploaded and is now available in the main Github page, here. We also added the following text in the supplementary to better describe the content of that file:

This file is a tab-separated list of every protein generated for the "Palaeoproteomic hominid reference dataset". Each protein is linked to at least one publication, where it was identified in either bone or tooth tissue. If multiple publications mention a protein, the names of these publications are all present and separated with a comma (' , ').

Supplementary Final Execution: This section feels incomplete. I'm not sure what is needed, but more detail is probably helpful.

We have updated the supplementary to provide more details on this final step of the generation of the Reference Dataset. The following text has been added:

~~Both BAM files and VCF files were then used as input for Module 2, as exemplified by the Tutorial.~~

Once the BAM and VCF files were prepared, they were utilized as input for Module 2 and the targeted proteins were translated for every chosen individual. The process of translating these datasets is described in detail in the github tutorial, in "STEP 2", including the preprocessing that some of the VCF files need to go through. We used GRCh38 as the annotation reference for datasets 1,2 and 3 and GRCh37 for datasets 4 and 5. The generated protein fasta files were collected from the output folder and assembled into the final folder, which is available in Zenodo.

Reviewer - Katerina Douka

This is an exciting development in the field of palaeoproteomics and one that the community will welcome. I recomment the manuscript for publication and include below my comments and some minor corrections/additions.

1/ To maket the manuscript appear more informed, I would add in the first paragraph that while shotgun proteomics is used to infer phylogenic relationships, another palaeoproteomics method (PMF or ZooMS when for collagen) is used as a primary tool for identifying new hominid remains, which can then be analysed deeper with shotgun proteomics, ultimately using the new bioinformatics tool presented here.

We thank the reviewer for their positive comments and feedback. We agree that clarifying between shotgun proteomics, which is the main focus of the workflows presented here, and PMF approaches will help the reader better understand the scope of our work. To do so, we have added the following text and citations to the manuscript:

These ancient proteins can be utilized by Peptide Mass Fingerprinting (PMF) methods (Ostrom et al. 2000), including ZooMS (Buckley 2009 et al.), for genus or species identification (Buckley et al. 2010) and to single out fossil material of interest for further analyses including DNA sequencing (Brown et al. 2016, Brown et al 2022), radiocarbon dating (Deviese et al. 2017) and shotgun proteomics (Brown et al. 2016, Welker et al. 2016). Shotgun proteomics in particular, utilizing tandem mass spectrometry, has enabled the reconstruction of the amino acid sequences of those proteins, which sometimes number in the hundreds (Cappellini et al 2012, Warinner et al. 2014). ~~~ ~~The sequences of these proteins~~ These sequences contain evolutionary information…

If so, I would add a few more references aside from the Copenhagen group. We are talking about democratisation of the field, citing more widely is part of it too. I believe the oldest collagen analysed so far is presented in Rybczynski et al. (2013) also more recently expanded (Buckley et al. 2020, cited already), and other teams have also published very ancient proteins (e.g Nielsen-Marsh et al. 2009 (https://www.sciencedirect.com/science/article/pii/S0305440309001253), or Brown et al. 2022 (https://www.nature.com/articles/s41559-021-01581-2).

We agree with the comment of the reviewer that the initial citations unintentionally overrepresented certain palaeoproteomic groups over others. We have tried to amend this both with the text added in the previous comment, as well as by citing the suggested literature and some additional works from other groups:

Additional citations in main text: Nielsen-Marsh 2009, Rybczynski 2013, Nogueira et al. 2021

2/ Page 2. "The amount of publicly available proteome sequences is much smaller in comparison".-> Can you quantify this? There are indeed very few.

Quantifying this accurately might be a difficult task since different databases will provide different numbers. We made a rough estimation using two of the largest databases: NCBI's "Genome" for whole genomes and Uniprot's "Reference proteomes" for full proteomes. These 2 databases might not have the same standards for every entry they contain (e.g. in Uniprot the Homo sapiens proteome has 82,400 entries while the Diceros bicornis has 19,600), but should provide an estimate of how many different species are available. Our results from this search are presented below and will be added in the main text:

NCBI's list of sequenced genomes (www.ncbi.nlm.nih.gov/genome/) includes 78,420 species, out of which 30,530 are eukaryotes and 11,345 labeled as 'Animal'. For comparison Uniprot's reference proteomes list (ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/reference_proteomes/README) contains a total of 23,805 entries of which 2,400 are eukaryotes and around 950 'Metazoa'. Finally, Ensemble's database of fully annotated genomes (https://www.ensembl.org/info/about/species.html) , and thus available proteomes, numbers to only around 270.

3/ For Module 3, I would have appreciated comments on thresholds or limitations for the use of PaleoProPhyler. Are there are any? What are the limitations imposed by the (often) small number and of poor preservation of proteins/peptides for a given sample. Are there cut-offs and suggestions how to overcome them?

This is indeed an important subject of the field that often gets overlooked. Protein data are less variable than DNA due to synonymous codons as well as selection acting upon them. In addition to this, ancient proteins are usually few in number and are characterized by their lower quality, especially in terms of missingness. All these aspects combined with biological phenomena such as admixture and incomplete lineage sorting can lead to complications in inference of species relationships. However, we wanted to refrain from defining any thresholds or cutoffs, as the workflow itself does not have any functional constraints or built-in cutoffs per se.
The effects of the aforementioned issues will require an investigation of their own and will likely differ between taxonomic levels (e.g. comparing species of the same family vs species of the same genus) and different taxonomic groups of the same level (e.g. comparing species within Hominidae vs comparing species within Elephantidae). Instead, we now mention potential issues, so that the users of the workflow are aware of them. The workflow does incorporate some of the most commonly employed phylogenetic software which themselves generate trees with confidence metrics such as posterior probabilities or bootstrap values. Although these metrics can in rare cases provide falsely high confidence, they can serve as a test for the quality of the phylogenetic interpretations given the input data. Finally the workflow also provides the user with the ability to 'mask' a modern reference sample with the missingness of an ancient sample. This

allows the user to get a rough estimate of how much the missingness of their ancient sample could affect its phylogenetic placement. We have added the following to the main text, in a small paragraph named 'Closing Remarks':

The workflows presented here aim to facilitate phylogenetic reconstruction using ancient protein data to a wider audience, as well as to streamline these processes and enable greater reproducibility in the field. Although we highly encourage the use of the tools and methods utilized by our workflows, we still caution against the overinterpretation of paleoproteomic results. Deriving species relationships from ancient proteins is still a relatively new endeavor and as a result, our understanding of this data, their quantity and quality requirements, robustness and accuracy are all largely unexplored. We believe that palaeoproteomic data should therefore be used in combination with other sources of information in order to make accurate evolutionary inferences.

4/ There is a mention for Supplementray Material, I could not see it or access it.

*The **original** pdf file should be available in the main biorxiv page (top right link - 'Supplementary Material') of the preprint: https://www.biorxiv.org/content/10.1101/2022.12.12.519721v1 . Additionally, the **latest** version also be accessed in the github page: https://github.com/johnpatramanis/Proteomic_Pipeline/blob/main/GitHub_Tutorial/Supplementary.pdf*

5/ Unless there is a very specific word limitation, there is very little in the description of how the pipeline works and even what each Module does. I like the graphical abstract but I was left wondering where is the input and output and, as alaready mentioned, indication of cut-offs and generaly data hygene.

The initial submission of the manuscript aimed at a minimal and concise format. Given that all reviewers have requested more information on the applications of each module, we have decided to move some of that information from the supplementary into the main text. We have added the following text into the manuscript:

Module 1 is designed to provide the user with a baseline (curated) reference dataset as well as the resources required to perform the *in silico* translation of proteins from mapped whole genomes. The input of module 1 is a user-provided list of proteins and a list of organisms. The user also has the option of choosing a particular reference build. Utilizing the Ensembl API [1], the module will return 3 different resources for each requested protein and for each requested organism. These are: a) the reference protein sequence of that organism in FASTA format [2], b) the location (position and strand) of the gene that corresponds to that protein and c) the start and end of each exon and intron of that gene / isoform. The downloaded FASTA sequences are available individually but are also assembled into species- and protein-specific datasets. They can be immediately used as a

reference dataset for either downstream phylogenetic analyses or as an input database for mass spectrometry software, like MaxQuant [3], Pfind [4], PEAKS [5] and others [6, 7, 8, 9]. The gene location information and the exon / intron tables can be utilized automatically by Module 2. For each requested protein, the module will select the Ensembl canonical isoform by default. Should the user desire a specific isoform or all protein coding isoforms of a protein, they have the ability to specify that as an option in the provided protein list.

Module 2 is designed to utilize the resources generated by Module 1 and to extract, splice and translate genes from whole genome data, into the proteins of interest. This module can handle some of the most commonly used genomic data file formats, including the BAM [10], CRAM [11] and VCF [12] formats. The easiest way to run Module 2 is to first run Module 1 for a set of proteins and a selected organism. This will generate all the necessary files and resources required for the protein translation. This selected organism will be then used as a reference for the translation process. All genomic data to be translated in a single run must be mapped onto that same reference organism. The user can then run Module 2 simply by providing the organism's name (and optionally a reference version), as well as a list of the samples to be translated. The user can also translate samples from a VCF file, but they will need to provide a reference genome in FASTA format, to complement the variation-only information of the VCF file. The translated protein sequences are available individually but are also assembled into individual- and protein-specific datasets.

Module 3 is designed to perform a phylogenetic analysis, with some modifications needed when working with palaeo-proteomic data. The input of this module is a FASTA file, containing all of the protein sequences from both the reference dataset and the ancient sample(s) to be analyzed. The dataset is automatically split into protein specific sub-datasets, each of which will be aligned and checked for SAPs. The alignment is a two step process which includes first isolating and aligning the modern/reference dataset and then aligning the ancient samples onto the modern ones using Mafft [14]. Isobaric amino acids that cannot be distinguished from each other by some mass spectrometers are corrected to ensure the downstream phylogenetic analysis can proceed without issues. Specifically, any time an Isoleucine (I) or a Leucine (L) is identified in the alignment, all of the modern sequences are checked for that position. If all of them share one of the 2 amino acids, then the ancient samples are also switched to that amino acid. If both I and L appear on some present-day samples, both present-day and ancient samples are switched to an L. The user also has the option to provide an additional file named 'MASKED'. Using this optional file, the user can mask a present-day sample such that it has the same missing sites as an ancient sample. Finally a small report is generated for each ancient sample in the dataset, and a maximum likelihood phylogenetic tree is generated for each protein sub-dataset through PhyML [15]. All protein alignments are then also merged together into a concatenated dataset. The concatenated dataset is used to generate a maximum-likelihood species tree [16] through PhyML and a Bayesian species tree [17, 18]

through MrBayes [19] or RevBayes [20]. The tree generation is parallelized using Mpirun [21].

Some minor stuff:
"...lab-generated protein data does not even exist" : Remove even
"...absence of knowledge about even a single amino acid polymorphism": Remove even
"The modules are intended to synergize with each other" : I am not sure of the word synergize here. Maybe best to keep it simple and say "work with each other"

We thank the reviewer for the suggested corrections, which we have implemented into the text.

Anonymous Reviewer 2

The manuscript 'PaleoProPhyler: a reproducible pipeline for phylogenetic inference using ancient proteins' by Patramanis and colleagues presents an open-source pipeline for the phylogenetic analysis of palaeoproteomic data. The pipeline is split into three modules which follow on from each other, but can be run independently. These build a basic reference database from proteomes available on Ensembl (module 1), transcribe published genomes to supplement the reference database (module 2), and perform phylogenetic analysis of proteomic data using the reference database (module 3). The motivation for the development of the pipeline and a brief overview are provided in the main text with a more detailed explanation of the workflow presented in the supplementary information and the code available on the github of the lead author. A tutorial is provided to train users in how to install and run the pipeline using published data to reconstruct the enamel phylogeny of two hominids, Homo antecessor (Welker et al 2020) and Gigantopithecus blacki (Welker et al 2019). The authors used modules 1 and 2 of the pipeline to curate a hominid palaeoproteomics reference database which they make publicly available on Zenodo.

Open-source tools for reproducible data processing and analysis between different research groups and labs are important areas of development for the field of palaeoproteomics, as they are currently lacking. This hinders data reproducibility and represents a barrier to researchers within the field who lack formal training in computational biology. The PaleoProPhyler pipeline presented by the authors addresses this issue and therefore has the potential to be a timely and important addition to the toolset available to the palaeoproteomics community. The rationale for the work is clear and the manuscript is well written. The modularity of the pipeline is highly useful and will enable users to adopt portions of the pipeline for their own uses. The tutorial written with a 'non-bioinformatics-background audience in mind' is an excellent resource to increase accessibility to a wide range of researchers and achieve the aim of improving reproducibility within the field.

We thank the reviewer for their thorough read and comments on the manuscript.

I am not a bioinformatician so will not comment on the scripts themselves but will comment from the point of view of the 'non-bioinformatics' audience, the target audience of the tutorial. Unfortunately, I was only able to run the first module of the pipeline when following the tutorial, whilst Modules 2 and 3 resulted in errors and termination of the script. Perhaps readers with bioinformatics training would be able to adapt the scripts to make them run but even with access to server and bioinformatics support I was unable to complete the tutorial. Therefore, to be widely employed by researchers with different computational setups, some revisions to minimise dependencies and the potential for clashes between systems would be beneficial.

We thank the reviewer deeply for taking the time and effort to follow the tutorial and attempt to run each module of the pipeline. We are sorry to hear that they were unable to run modules 2 and 3. We are happy to help solve the errors encountered, if they could be sent to us (anonymously through here or via email or github, as preferred). Our goal is to make the tutorial as easy to run as possible, and

we expect it to evolve as we receive feedback from users about errors or other problems they might encounter.

The tutorial first directs the user to download the github workflow and install the conda environment from the command line and then download the published fasta files of the two hominid proteomes. As noted by the authors, the user ideally needs access to a high performance lab server for sufficient computational power to run the pipeline. The installation of the conda environments and pipeline may clash with pre-installed software on the institutional server which the user has no access to modify. This may act as a barrier to the installation of the pipeline.

We agree with the reviewer's comments on the installation of the workflows and have included alternative ways to install the required software of the pipeline. These alternative commands install only the bare minimum tools that are required for each module to work. They are available at the beginning of the tutorial in a new subsection called "Installation errors / Alternative Installations". We hope this will help users who may have a problem with the standard way of installing the software through pre-packaged conda environments.

The first module generates a scaffold reference database by downloading proteomes from species closely related to the hominids from Ensembl, a publicly available database for annotated genomic data. The second module is designed to supplement the scaffold reference database through the transcription of published genomic data, including other ancient hominins.

Running the first module was relatively quick and straightforward. Some further information or references could be added on the strengths/weaknesses of downloading reference proteomes from Ensembl vs translating genomes. I was unable to run the second module so cannot comment on the output.

We agree with this comment from the reviewer and have added text in the tutorial briefly explaining the strengths and weaknesses of using reference proteomes vs translating multiple genomes. This text has been added at the beginning of " Step 1" in the tutorial and contains the following:

Reference proteomes, like reference genomes, serve as the 'default' representative of a given species. They can be used to easily compare organisms with each other and, in our case, with a new sample of unknown evolutionary placement. Although useful, they only represent a single organism from a given species or population. If one is interested in capturing a larger portion of the genetic diversity, and other genomes from that species or population are available, it is also possible to use PaleoProPhyler to translate multiple user-provided genomes into proteomes, using Module2.

The third module merges together the palaeoproteomic data with the reference datasets and performs phylogenetic analysis. Implementing the module seems very straight-forward, however the

tutorial ends abruptly after the analysis has been run with no further information on where the output files are generated. The tutorial could be improved by adding additional information here on how to check the output of the analysis (as the authors did at the end of module 1), how to visualise the trees generated data and some simple QC checks to carry out.

We thank the reviewer for their helping comments. We have now added additional information at the end of 'STEP 3' of the tutorial, including guidance on how to visualize the resulting alignments and phylogenetic trees, and how to evaluate the quality of those trees.

Although the pipeline may run successfully on the author's institutional server, it needs to be packaged more efficiently for widespread use. There appears to be some typos in the code or system incompatibility which prevent the pipeline from running to completion. It would require a bioinformatician to troubleshoot the errors. This is therefore a barrier to anyone without this knowledge base.

This is a common problem when sending code between labs and can require some complicated trial and error to solve. I suggest packaging the software into a container so it can be shared between labs without issues of installation in clashing systems. Enlisting several researchers from different labs outside the Globe Institute to install and run the pipeline tutorial on their own servers would provide the authors with the opportunity to trouble-shoot any issues that arise.

We thank the reviewer for their comments and suggestions. To replicate the problems raised by the reviewer, we reached out to 2 additional groups outside of the Globe Institute (as well as one inside) to test the pipeline and troubleshoot errors as they rise up. We also tested the workflow ourselves by receiving access on two different servers that utilize a popular queueing system ("Slurm").

During this process, we identified an issue with a specific software, namely 'MrBayes' , having conflicts in particular types of Slurm server environments. Essentially, the conda version of this software may be successfully installed in these environments, but raises an error when run (this does not happen when installing it manually). The tool in question has thus temporarily been turned off in the pipeline, by default. We have also submitted an official report for this error in the main github page of MrBayes: https://github.com/NBISweden/MrBayes/issues/283 . An alternative Bayesian phylogenetic tool (with even more phylogenetic capabilities), 'RevBayes', has instead been implemented into the pipeline as a replacement. Users can still run MrBayes if they wish so, by re-activating it as described in the main page of the github (https://github.com/johnpatramanis/Proteomic_Pipeline). Finally we have also added a new section at the end of the tutorial and the main github page titled "Reporting Issues", which encourages the reporting of errors and suggests a format for those reports.

Other points:

- *The system requirements for running the pipeline on a linux OS are not apparent until the SI and tutorial - this could be mentioned in the main text under 'Availability and Community Guidelines'.*
- *The hominid reference database will be highly useful. Although the references for the data are available in the SI, a table with all of the individuals included would be useful.*

- *Overall the authors have done a good job adding useful tips, warnings, additional descriptions and links to resources to help users who are new to this type of analysis. Perhaps a text box with a glossary/key terms to provide additional descriptions of the different file types (FASTA, VCF, BAM. CRAM) could be useful for a non-bioinformatics audience, as there are lots of abbreviations used.*
- *Ref 61 in the first paragraph of the Statement of Need appears to have no link.*
- *There are some typos throughout the tutorial text so some proofreading would be beneficial.*

We thank the reviewer for these additional points. We have done our best to address them as well:

- We have added the requirement of a Linux OS in the 'Availability and Community Guidelines' section.
- We have added a table in the main github page with all of the individuals present in the hominid reference dataset (https://github.com/johnpatramanis/Proteomic_Pipeline/blob/main/Reference_Sample_List.csv). This table, in the form of a tab-separated file, connects the name of each sample with the species or population (e.g. 'modern humans' vs 'neanderthal') it belongs to, the publication the data originates from and the type of data that was used for the translation of that sample (e.g. VCF or BAM files / ancient or modern DNA). We also included the link to this table in the Zenodo dataset description as well as the section of the SI that describes the creation of this dataset.
- We have added a small glossary at the very end of the tutorial with a small explanation for key terms of the document. It contains a quick explanation by the main author as well as a link to a more proper explanation.
- We have made an effort to clean up typos in the tutorial text.