We thank the reviewers for examining and commenting on the specification. We have addressed all the reviewers’ comments as follows in green:

Reviewer #1:

It looks like neXtProt has not implemented a recent version of the standard, as their PEFF files did not validate and contain several violations around the \Processed keyword.

As we were submitting the specification, neXtProt was testing their PEFF export against the submitted specification. With the latest neXtProt release available here (with file dates 2019-02-13):

ftp://ftp.nextprot.org/pub/current\_release/peff/

There are no validation errors. We apologize for the out of sync problems, but it is now resolved.

[I’m pretty sure this is right, but double check. Download and test again]

Beyond the individual notes, I think PEFF is a great idea, and it's looking more stable than a year ago. I don't think there are any fundamental issues remaining, just minor consistency checks and requests that the document be more explicit in some places.

[TO DO] The reviewer has made multiple minor edits and comments in the specification document itself. These have all been addressed with tracked changes and as noted here.

As an aside, I spoke with Eric Deutch about representing glycosylation patterns directly in the file format at ASMS last year, though we didn't reach any substantive conclusions. A non-trivial number of annotations were glycosylation related in the examples, but this standard pushes them off into the "free text uncontrolled" domain. I don't think this is the venue to campaign for their inclusion, though I don't know where specifically to do so. I just defended my thesis on glycoprotein identification algorithms, so I am biased towards their importance and integration into the informatics ecosystem.

The format is biased toward PSI-MOD and Unimod controlled vocabularies. PSI-MOD does already contain a substantial number of glycan that could be used already as is. However, to capture a more complete annotation of glycan complexity would require a relatively small extension to the format. None of the creators of the format are working directly with glycans, so we would welcome participation by someone who is interested in using PEFF for this purpose. We agree with the reviewer that this is not the time to design a glycan system for PEFF, but with willing additional expertise, we would be happy to design such a glycan extension for PEFF in the future.

Also aside, Pyteomics [1] was recently updated and published to include an implementation of PEFF parsing, based upon the implementation I wrote for my own library (https://github.com/mobiusklein/glycopeptidepy) which supports reading and writing PEFF.

[1] Levitsky, L. I., Klein, J., Ivanov, M. V., & Gorshkov, M. V. (2018). Pyteomics 4.0: five years of development of a Python proteomics framework. Journal of Proteome Research, 18, acs.jproteome.8b00717. <https://doi.org/10.1021/acs.jproteome.8b00717>

[TO DO] Outstanding! This is a welcome addition. We have added a note to the specification pointing to these Python for PEFF resources.

Reviewer #2

The manuscript describes a new file format to store protein sequences including valuable information about variants, post-translational modifications and other processing events, which are usually not captured by common FASTA formats.

Given the lack of a widely accepted standardized file format for protein sequences and their additional features, the PEFF database format will be of high value to the proteomics community. While the format is mostly well described, there are still several shortcomings which I outlined below. Therefore, I recommend revision of the manuscript.

We thank the reviewer for careful reading and insightful comments, which we address as described below in green.

*Currently available file/sequence formats:*

Earlier attempts to develop formats to describe proteoforms should shortly be acknowledged. For example: "Protein Ontology (PRO): enhancing and scaling up the representation of protein entities" and some work by H Schlüter (e.g. 10.1186/1752-153X-3-11).

[TO DO] We have added an acknowledgement and citation to the article mentioned above.

*Definition of a subformat with specific (exact?) proteoforms:*

This part is rather unclear as a description is missing of what the authors mean with a proteoform database. Must the proteoforms be uniquely defined (no ambiguity, only fixed amino acids, ...)? Then this will imply additional rules and restrictions such as for the amino acids and PTMs (multiple described at once). This is only remotely mentioned when describing the ModResUnimod key. For example, is it allowed to have a modification at multiple places described in one key value pair (e.g. (100,5075|UNIMOD:21|Phospho)) in a proteoform database?

In order to make the different types of databases clearer, why not further distinguish fixed and "flexible" proteoform databases? In the latter, single proteoforms with ambiguities in e.g. PTM position could be described.

I suggest to also mention the combinatorial explosion arising from the many combinations in the "open" database format. This format will then describe the space of possible and not actually known proteoforms. It seems that the purpose of such a file will be very different from e.g. reporting/collecting a "real" proteome.

[response TO DO. Not quite sure what to respond here.]

*Features that seem to be missing:*

How to describe N/C-terminal modifications?

[Do we need something special here? Or is this just a mass modification on the terminal AA?]

What about cross-links in general?

[We have addressed disulfide bonds specifically. What about other kinds of cross-links? I am aware of artificially added cross linkers. We could think up a way to encode those. Or are there other natural ones we should consider?]

Is there a reason to avoid the RESID database for PTMs?

The reason for avoiding RESID is that PSI-MOD contain all (nearly all?) of the relevant entries in RESID, and supporting yet another reference CV for modifications would make the task of reading PEFF files another step more complex. I would be very easy to add support for \ResidMod=(5|RESID:123|modName) either as a custom addition or with official support in the CV. But is useful and advisable or just a needless another way to do it?

What about new proteoforms that are not part of any database, should they be described through a custom database? Is there a naming procedure for that, i.e. to provide Prefix, DbVersion, DbSource, ...?

If new proteoforms being described are not present in any other source, then the PEFF file becomes the originating source, and it would be advisable and required to self-appoint a prefix, version, and source for the collection of new sequences. We have added some text to the specification describing this [TO DO].

*Further comments:*

1.2 What do the authors mean with exact proteoforms? This has to be related with the

ProteoformDB tag. There, they write "specific proteoforms". Definitions of the terms will help.

[This is probably some careless language intended to disambiguate the case where each entry is a general protein sequence which can describe many proteoforms from cases where each entry is exactly one proteoform. We have made this clearer [somehow. TO DO]

3.3.3: Proper handling of parenthesis requires them to be in the form "(...)". Only an opening or

closing parenthesis will cause trouble.

This is true. It seems easier to disallow such diabolical cases rather than support them. Is there a reasonable use case for an unpaired parenthesis? If so, this would require introduction of character escaping functionality. For now, we have added a statement disallowing unpaired parentheses. [TO DO]

3.4.1 mentions the ProForma nomenclature and the authors are rather optimistic about the

conversion between the formats. For instance, disulfide bonds can be used in ProForma as its

description proposes a way to describe cross-links. I also doubt that translation from ProForma to

PEFF does occur without loss of information. The following items are at least very difficult to

translate: mass shifts, sequences not characterized in a database and chemical formulae. Instead,

the authors should emphasize the differences of the purpose when comparing ProForma with PEFF.

ProForma is meant to describe actually measured proteoforms and does not relate to annotated

databases such as UniProt. On the other hand, PEFF is useful to describe possible proteoforms and

handle or even merge databases with details about the proteins not captured so far.

We thank the reviewer for these insightful distinctions. We have updated these text of 3.4.1 to reflect these sentiments. [TO DO]