Response to SC1 (Editor Johannes Wagner):

We thank the Editor for examining the database serializations from a user point of view and providing us with this helpful feedback. Below, we detail the revisions we will undertake in order to clarify usage of the serializations, and to provide additional sample code for non-R users. The original comment is copied below in **bold text** and our responses appear in plain text.

- Bronwen Konecky
On behalf of all authors

I think you present a well prepared product worth publishing, however looking through the data I noticed a few points that could be clarified before publication.

- if I look at the 3 serialisations, I see that the R and matlab versions contain d, TS and sTS, however the python pickle only contains D and TS
>>> infile = open('iso2k0_14_2.pkl','rb')

>>> iso2k_dict = pickle.load(infile)
>>> iso2k_dict.keys()
dict_keys(['D', 'TS'])
does that have any relevance? please clarify

- directly regarding those: I have not found in the paper what D TS and sTS stand for. For the undiscerning user, this is confusing.

We thank the Reviewer for this very helpful observation; the three variables are indeed different and require explanation. We have therefore added the following short paragraph in section 6.2 (between lines 685-690 of the Discussion paper) describing the D, TS, and sTS variables. The text is below.

The MATLAB and R serializations contain three variables: 'D', 'TS', and 'sTS.' The variable 'D' includes site-level data for each dataset structured in the LiPD format. Datasets in 'D' often contain multiple variables (e.g. stable isotope, ancillary, and chronological data), and represent how LiPD data appear when loaded into the initial environment. For most users, however, a "flattened" version of the database is more useful. We have provided this as the 'TS' variable, where each entry contains an individual time series and its associated metadata. A slightly modified version of 'TS' is included with R and Matlab, called 'sTS', which is identical to TS except that the interpretation fields are split by scope ('isotope' or 'climate') in order to simplify querying, which may be preferable for some users. The Python serialization contains only 'D' and 'TS'.

- your dataset at figshare makes no mention of the ESSD paper and the data are not well described at the repository. Whoever stumbles uppon your data there will be at a loss. Could you add a little data description or a readme file to the repository? The

very good metadata spreadsheet you provide is also missing there.

Thank you for this suggestion. Text will be added to the Figshare page with a link to the ESSD paper where users can find a description of the database, metadata fields, and supplementary tables. Links will also be added to download the 3 sample codes.

NOAA-WDS is in the process of getting a DOI minted for Iso2k version 1.0.0 (the "final publication" and official release of the database). It is our intention for that NOAA-minted DOI to be the DOI that is published in the final ESSD publication, rather than the Figshare DOI. Recognizing that v.0.14.7 of the database will be available via the Figshare DOI in perpetuity, a note will be added to that Figshare page directing users to Iso2k v1.0.0 and all future updates via the NOAA-minted DOI, and the NOAA-WDS landing page.

- lastly, for final publication, it would be very helpfull to provide a short sample code for the other two serialisations as well (in a free matlab implementation like octave, and python). I don't think they would have to be as extensive as the R sample. (This may also resolve issue number one, maybe you handle the m files differently)

We have created MATLAB and Python sample codes that replicate the example workflow in the original R sample code. The 3 sample codes are now available in a revised Supplementary Materials for this paper. The variable handling with Python is indeed a little different than with R and MATLAB, and the sample code now makes those procedures explicit. Thank you for this helpful suggestion!