General remarks

The paper seems long. The length, and difficulty of feeling out the structure of a long paper, probably led to a feeling of confusion about whether I was reading a geological case study, a paper on a new approach to ingesting data, or the documentation for a software package. Perhaps the manuscript is trying to do too much? I wonder if you could split it into pieces? For example, move the more prosaic stuff (lists of tables, etc) to the docs; describe the more technical language modeling piece in a short nerdy paper about that; and present the case study as a short geological success story?

→ We agree that the paper is quite lengthy. However, splitting the content into multiple papers may not be as impactful. We addressed the sense of confusion by restructuring the paper into two main sections, the theory and the case study. The structure is as follows:

1. Introduction
2. dh2loop Drillhole Data Extraction

2.1 Conventions and Terminologies
2.2 Dependencies
2.3 Data Source
2.4 Thesauri
2.4.1 Drill Hole Lithology Codes Thesaurus
2.4.2 Clean-up Dictionary
2.4.3 Lithology Hierarchical Thesaurus
2.5 Data Extraction
2.5.1 Collar Extraction
2.5.2 Survey Extraction
2.5.3 Lithology Extraction
2.6 Fuzzy String Matching Assessment

3. Case Study: Yalgoo-Singleton Greenstone Belt

3.1 Study Area
3.2 Data Extraction Results
3.2.1 Collar
3.2.2 Survey
3.2.3 Lithology: Lithology Code Workflow and Comments Workflow
3.3 Fuzzy String Matching Results
3.3.1 Structure and Texture
3.3.2 Igneous Rocks
3.3.3 Sedimentary Rocks
3.3.4 Metamorphic Rocks
3.3.5 Surficial Rocks

4. Discussion

4.1 dh2loop Functions and Notebooks
4.2 Thesauri
4.3 Data Extraction
4.4 Assessment of String Matching Results

4.5 Value of the Lithological Information Extracted for Multiscale Analyses
5. Conclusions

If the authors and editors feel a long paper is justified, then my suggestion is to be extra careful about the outline of the paper, so that a person knows what aspect each chunk of the paper is addressing.
⟶We agree. Please see previous response.

Compounding the length is that there's quite a bit of redundancy between text, figures and captions. Figure 9 is an archetypal example of this. I think the caption (which I cannot parse) essentially spells out the diagram (which I cannot read very well), and virtually all of the data is also written out in sections 3.1 to 3.3. My advice is to frame the ideas with the text and put all the details of table names and data in the figure, which should barely need a caption.
⟶We agree. We removed most of the redundant text and kept the information in the captions.

Some pieces seem better suited to the tool's documentation. Appendices B and C probably don't need to be part of a paper. Similarly, I think detailed descriptions of tables are clogging up the text and getting in the way of the reader. Standard measures like precision and recall, or string matching statistics, probably don't need to be described in detail.
⟶We addressed this and removed it from the Appendices. The standard measures were also removed.

Detailed remarks

- Title: Python should be written with an uppercase P.
  ⟶This was revised in the updated version.
- Abstract: I'm not sure you need the first paragraph; it's introductory material. If you want to motivate the problem, I think you can do it in a sentence (the second sentence captures it for me).
  ⟶This was revised in the updated version.
- Line 12: surveys, textual (no 'r')
  ⟶This was revised in the updated version.
- Line 15: hundreds of
  ⟶This was revised in the updated version.
- Line 19: replace 'that provides the functionality to extract…' with 'for extracting…'
  ⟶This was revised in the updated version.
- Line 23: 86%
  ⟶This was revised in the updated version.
- Line 51: You want spaces after all your semi-colons.
  ⟶This was revised in the updated version.
- Section 2: I suggest sticking to sentence case for your subheadings; at least make them all the same.

→We made them all the same.

- Section 2.1, Conventions: The font is called Lucida, not Lucinda. I like the idea of a font convention, but if you're going to use one it is important to be consistent... and being consistent pretty hard. E.g. line 174: collar should be a table name, I think? Line 217: are these tables? Line 225: dh2loop should be italic.
  →We double checked the font convention to make sure none of it is missed.
- Lines 166–184: This paragraph is really hard to parse. I think the reader may start wondering if they need to know this information. If the info is important, I think you can let Figure 2 do the work.

→This was removed from the text and kept in Fig 2.

- Line 280–295: Similarly, I think you're just describing Figure 4.

→This was revised in the updated version.

- Sections 2.4.1, 2.4.2, 2.4.3: This feels like documentation.

→These sections are kept but reduces in detail.

- Lines 419–440: There is a lot of information here; I feel it is perhaps best suited to documentation.
  →We decided to keep this section as it explains the string matching algorithm used by dh2loop which influences its results.
- Table 1: Not sure what to make of this. What do the checks and x's mean? What is the score? Why is one row bold?

→This was revised in the updated version.

- Sections 3.1, 3.2, 3.3: See my earlier remarks about these paragraphs vs Figure 9 and its caption.

→This was revised in the updated version.

- Table 2 and Figure 10: I find these hard to get insight from.

→We decided to keep these.

- Confusion matrices: There are a great many data tables here but I am unsure what to do with this information. I see that there are a lot of under-represented labels (low support). The only reference I could find to this problem mentions normalizing the accuracy, but this doesn't really solve the problem, it just makes the confusion matrix colourbar fit better. So you're still unable to balance precision and recall (with small support, one of them is probably going to be bad). I'm not sure what you could do about it, other than get more data, but it might be worth mentioning.
  → We expounded on this. However, this is the nature of using real-word examples. Geologic logs have always been imbalanced as more detail is given to particular lithologies depending on the purpose of the study.

Remarks about code

There are some problems with the licences on some of the code you are using. **QDriller** (https://github.com/valheran/QDriller) is licenced under the GPL, which is a copyleft licence. This means that if you use it then your entire project must be licenced under the same (or a compatible) licence. **You are not allowed to use GPL'd code under an MIT licence.** So you have a few options:

1. If you keep that code as-is, you must licence your entire project under the GPL.
2. Ask the original author to grant you permission to re-use the classes you need under a more permissive licence. (Ideally, it would be, say, LGPL and a completely separate library that you could simply depend on, without copy/pasting the code across.)
3. Rewrite this functionality from scratch.

⟶This is not used in dh2loop and thus was removed from the repository.

GeoVectoLitho (https://github.com/IFuentesSR/GeoVectoLitho) is not licenced, as far as I can tell. Normally that would mean that you cannot use it without written permission granting you some rights. So make sure you have written permission to place it under your MIT licence, or you could ask the authors if they will consider an open licence. At the very least, you need to make it clear that the mlp.py module is based on work that is copyright of those authors and explain the terms under which you're using it. You should probably also exclude it from the MIT licence so that nobody comes along and picks ut up thinking it's FOSS.
⟶The mlp.py code was used to compare the results with MLP. This has been removed from the repository.

In each of these cases, if you cannot find a way to
The Where to start instructions on the repo need VTK, folium and ipyleaflet adding to the installs. You could just tell people to install everything in requirements.txt with pip install -r requirements.txt. After that the notebooks ran for me (though I didn't run them all right through).
In Notebook 0, the map did not appear; I didn't try to figure it out. Enabling the extension didn't help. (I'm on a Mac, Chrome browser.)
⟶We revised this.

I'm not sure these comments should form part of a peer review. If you don't plan to maintain this library into the future, you can probably just ignore all this. But in case it's useful:

- I strongly recommend writing docstrings for all of your functions.
- You should write tests for your functions. "Untested code is broken code."
- You can remove components of the standard library — sys, math, re, time, itertools, etc — from requirements.txt. These are included in every Python installation.
- I advise against hard-coding the file encoding in your functions (with encoding = "ISO-8859-1") because this seems like something another person is quite likely to want to change, e.g. if they have a UTF-8 encoded file. You should expose it as an argument. The same goes for database connections and other things another user might want to change.
- You should delete unused code, rather than commenting it.

- I'm sure you know about geopandas already, but it seems like it would be useful in your project.
- Note that PyProj emits a warning in Python 3.8+ FutureWarning: '+init=:' syntax is deprecated.
- I don't know if psycopg2 connections or cursors are compatible with Python's context manager (in which case you should 'with' blocks for them), but even if they don't you should put them in try-finally blocks to ensure they get closed no matter what happens in the runtime.
- I recommend using a linter like flake8 to find things like multiple or redundant imports (e.g. you import numpy three times in dh2l_db.py). It will also help you adhere more closely to the PEP8 standard, which will make your code more readable and reusable.

Overall, the code has a non-Python feel to it (I don't know Fortran, but I've read a lot of code from geophysicists and they often write code like this!). Patterns like functions mutating global variables are not common in Python. Typically, one would pass the variable in to the function, then return them to the user (these are so-called 'pure functions'). You will find this easier to maintain. It feels strange to me to run a function like dh2l.litho_dico(litho_dic_file) and not have something (like a new DataFrame) returned to me.
→We will keep the code in current form and address these suggestions in a future release.