Supplement of

OneDZ: A Global Detrital Zircon Database and Implications for Constructing Giant Geoscience Database

Keran Li et al.

Correspondence to: Xiumian Hu (huxm@nju.edu.com)

Outline

1 AI tools in constructing database

The web address for DataExpo is http://dataexpo.deep-time.org/. The main website is shown in Figure A1. By using "detrital zircon" as the keyword, the system can find and rank relevant items on open-access websites to identify potential online databases (Figure A2).

Regarding the use of GPT as an agent to find papers, it's important to note that this involves a workflow for processing data. Large language models (LLMs) currently face limitations in directly searching specific websites like Google Scholar, Geoscienceworld, and others. Therefore, an essential step is to collect summaries of potential titles from these websites. In this research, we tested the GPT-Agent workflow on Google Scholar. Initially, we adapted a collector from a GitHub repository (https://github.com/JessyTsu1/google_scholar_spider). We then applied ChatGPT using the API key locally. After configuring the keys, GPT was used to adjust title information. Following this, GPT worked with the adjusted title tables, and through several prompt engineering cycles and interactive checks, it was able to predict the sources of zircon data with fair accuracy.

DeepShovel (https://deepshovel.deep-time.org/) is an online platform integrated with artificial intelligence technologies, specifically designed to assist researchers in Earth sciences with data extraction from scientific literature in PDF format (Figure A3). The platform leverages advanced neural network models and user interaction to efficiently identify and extract data from tables, figures, maps, and text within the documents.

To utilize DeepShovel, researchers initially upload literature files containing the desired data. The platform then automatically parses different sections of the documents, including metadata, text, tables, and maps. For metadata extraction, DeepShovel employs tools such as Grobid and Science Parse, integrating extraction results from various tools through a voting mechanism to obtain key information like the title, authors, abstract, publication year, etc., of the literature.

For extracting academic entities from the text, the platform uses weak supervision learning models and rules to assist users in identifying and extracting specific information, such as geological era names. Table data extraction is another critical function of DeepShovel. The platform employs object detection models like Detectron2, combined with rules and user interaction, to help users locate and recognize tabular data within the literature. Users can adjust the structure of the table with system assistance and utilize optical character recognition technology such as Tesseract to extract cell content.

Map recognition and geographical location extraction are also key modules of DeepShovel. The platform can recognize maps within the literature and assist users in determining the latitude and longitude of points on the map through drawing and marking operations. Finally, all extracted data can be integrated into a unified table during the data integration phase, facilitating the construction of a scientific database. DeepShovel supports project-level data integration, allowing users to set the header of the master table on the project page and automatically match and integrate data from various parts of the document.

The introduction of the online platform is detailed in Zhang et al. (2023), Zhang et al. (2022a), and Zhang et al. (2022b).

2 Python and MySQL code snippets in data process

In this research, we developed Python scripts primarily aimed at automatically checking and removing erroneous rows in the dataset. These scripts include DuplicateRemove.py for removing duplicate rows, DuplicateCheck.py for identifying duplicates, and DuplicateLog.py for generating logs. All scripts utilize the Pandas package to operate on CSV, XLSX, or SQL files. However, we do not recommend using XLSX files for data cleaning, as Excel tables with over 1,000,000 rows require at least 128GB of running memory. Additionally, we recommend a minimum of 64GB of RAM on personal computers when using these Python scripts to process the GDZ database.

To address the inefficiencies of Python scripts, we have moved away from using Excel files for database management. Excel files not only suffer from low efficiency but are also prone to formatting issues. While MySQL is a secure and efficient option for managing databases, its lack of a visual interface can be a barrier for Earth science researchers. To enhance user-friendliness, we introduced Navicat software, which provides a convenient visual interface for interacting with MySQL. Additionally, we have developed various MySQL scripts for tasks such as character modification, statistics, and data checking, ensuring fast and accurate data cleaning.

All the Python and SQL code snippets mentioned in this research are available for download at https://github.com/KeranLi/Global-Detrital-Zircon. This resource aims to provide researchers with the tools necessary to efficiently manage and clean large-scale datasets within the GDZ database.

3 Python code snippets in coordinates

Firstly, the automatic conversion process must account for the multiple formats of DMS (Degrees, Minutes, Seconds) coordinates. After careful re-evaluation, we identified that the primary DMS formats use either symbols like $\circ$/$\prime$/$\prime\prime$ or the terms "degree/minutes/seconds" to represent coordinates. Other more complex cases include combinations of symbols, letters, spaces, slashes, dashes, and varying capitalization. To handle this, we implemented a fuzzy retrieval method that first automatically detects the separator symbols. This allows for quick extraction of values from different time scales based on the identified separators, followed by rapid calculations.

Moreover, the construction of the database involves importing multiple DMS format data in batches for conversion. Given the variety of data storage formats generated during the crowdfunding process, we have also developed fast automated parsing of multivariate files within Python scripts, further enhancing the efficiency of data conversion.

In addition to coordinate conversion, we also implemented a method for estimating latitude and longitude coordinates based on geometric relationships in images, using Python. This method considers the distortion effects caused by projection and manual selection errors. Projection distortion is closely linked to the choice of projection mode during map drawing. In sedimentology, it is often challenging to trace the original author's settings for projection modes in spatial maps. Therefore, we determined the distortion parameters experimentally. First, known spatial points were projected using different projection modes at various spatial scales (e.g., 1 km by 1 km). Then, the estimated coordinates of the

target point in the image were repeatedly measured, and the distortion coefficient was iteratively calculated until the estimated coordinates closely matched the actual coordinates.

All the Python code snippets used in this research can be downloaded from https://github.com/KeranLi/Global-Detrital-Zircon. This repository provides the tools necessary for efficient data conversion and coordinate estimation in the context of global detrital zircon studies.

4 Paleo-globality method

The calculation of Paleo globality is further enhanced using the PyGplate package. Initially, the data with spatial coordinates is linked to a Plate ID. This bundled data can then be restored to any desired time using PyGplate's encapsulated code. The data points reconstructed based on rigid blocks are evaluated for globality using grid partitioning methods.

To improve code execution efficiency, we introduced GC packet dynamic memory management in this study. The computational processes were carried out on the Intel Xeon E5-2680 v3 (12 Cores, 30MB Cache, 2.50 GHz, 9.6 GT/s QPI) processor at the Supercomputing Center of Nanjing University, running on a Linux operating system.

All the Python code snippets used in this research can be accessed and downloaded from https://github.com/KeranLi/Global-Detrital-Zircon.

5 Figures

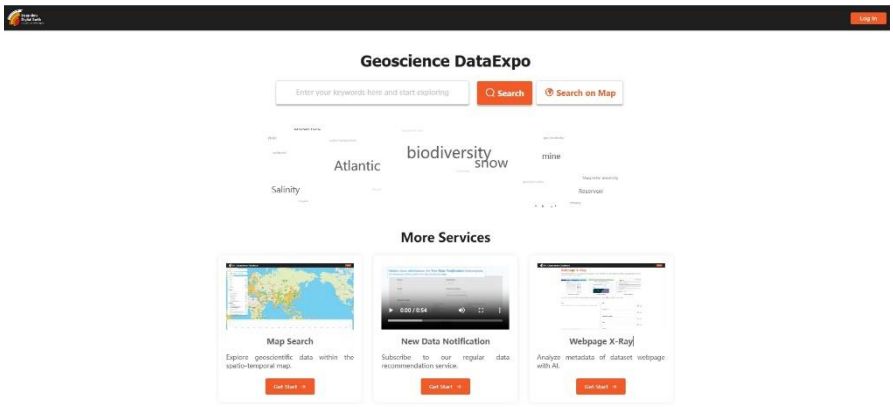

Figure S1: The homepage of DataExpo system.

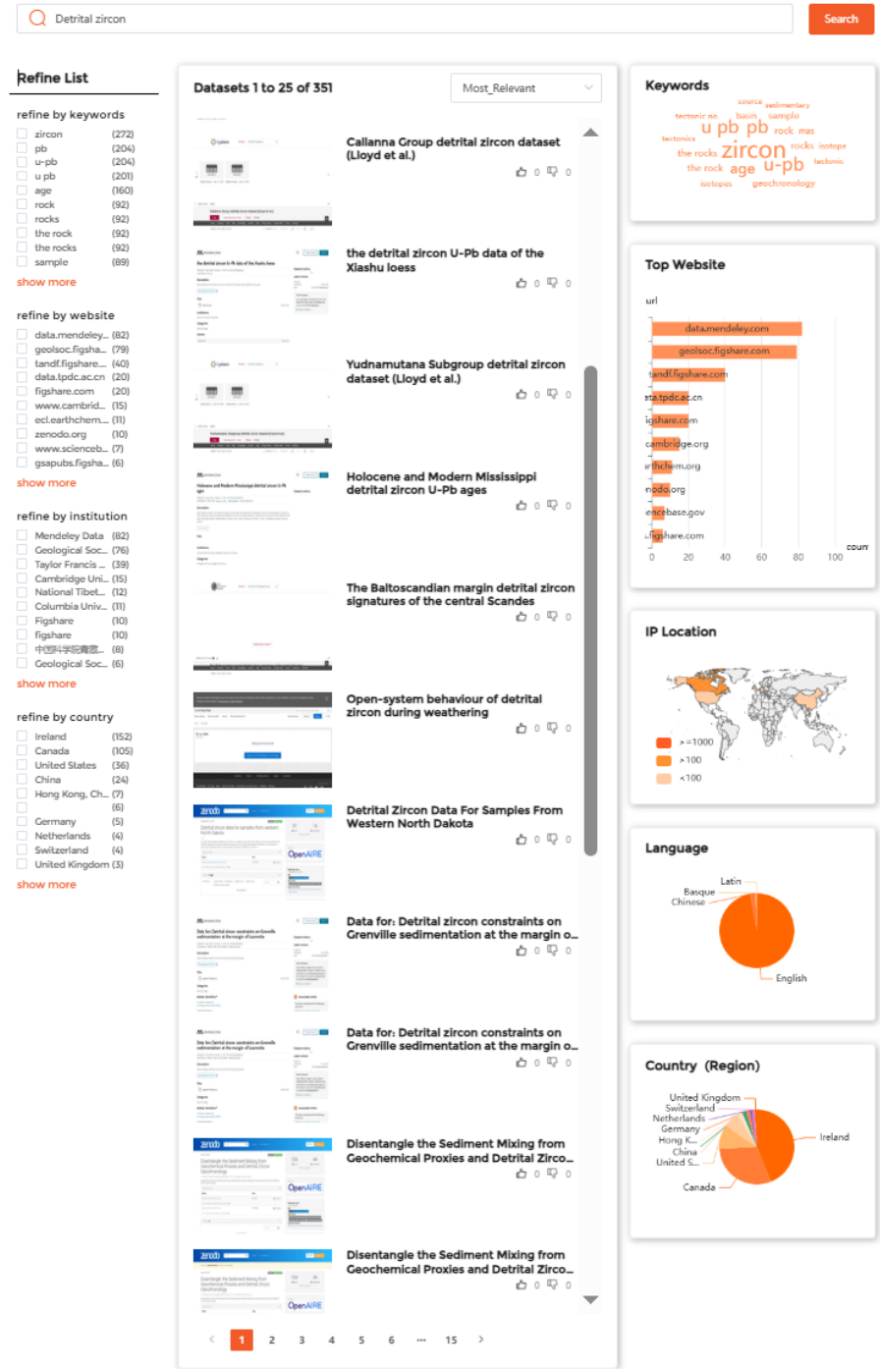

Figure S2: The search results of DataExpo system.

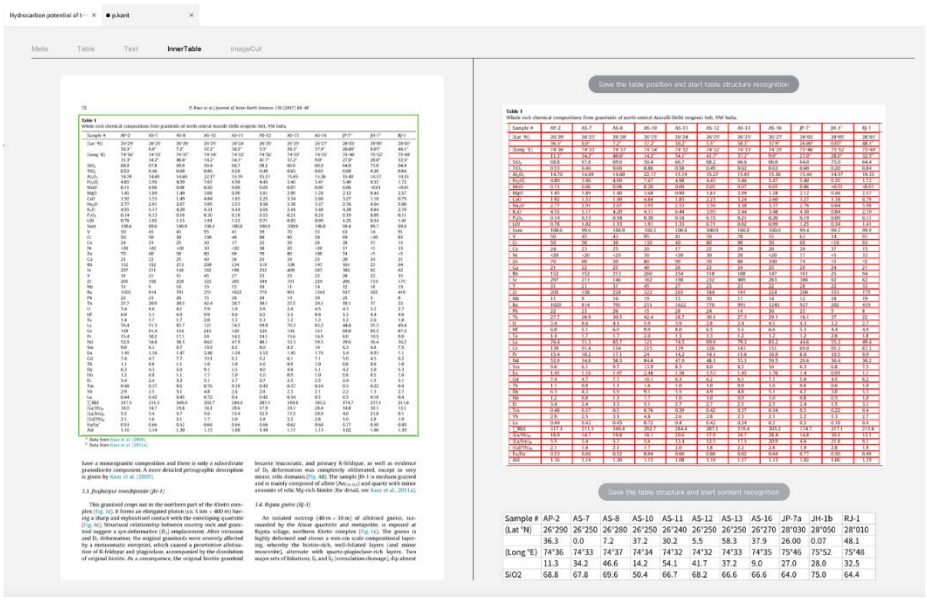

Figure S3: The interactive table extraction of DeepShovel.

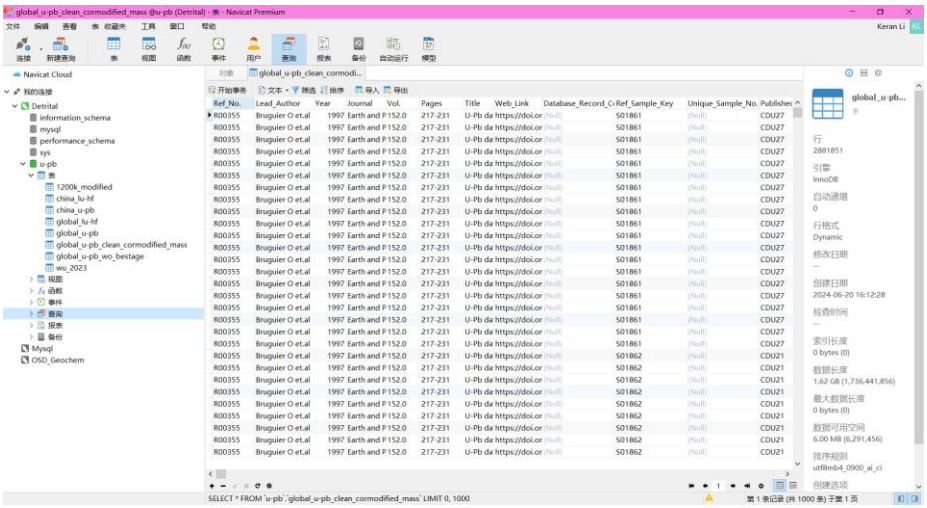

Figure S4: The GUI of Navicat.

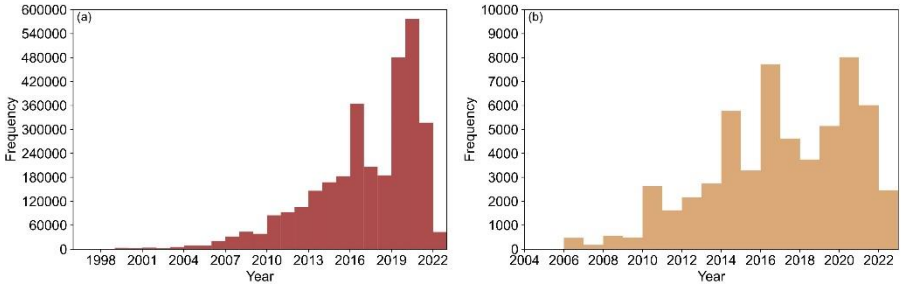

Figure S5: U-Th-Pb and Lu-Hf references variation with time. (a) U-Th-Pb; (b) Lu-Hf.

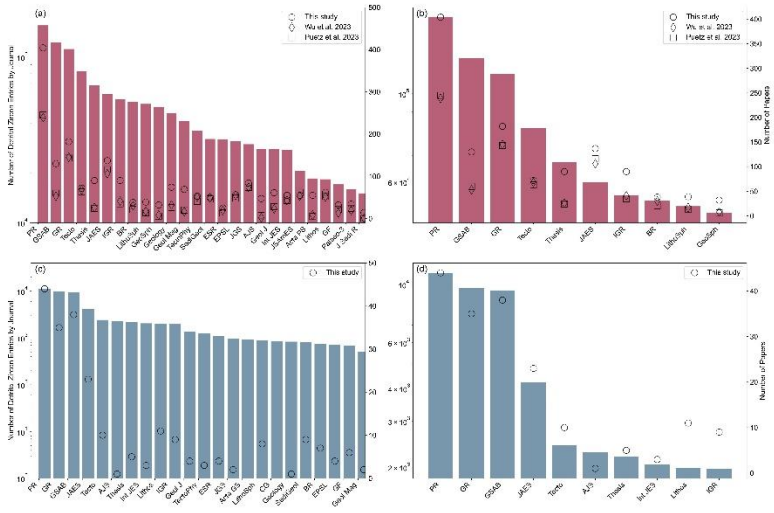

Figure S6: ar plots of the references. (a) The single zircons from different journals in U-Th-Pb data (Only display journals contribute zircons supreme 20000); (b) Top-10 journals contribute the detrital zircon records in U-Th-Pb data; (c) The single zircons from different journals in U-Th-Pb data (Only display journals contribute zircons supreme 20000); (d) Top-10 journals contribute the detrital zircon records in U-Th-Pb data. PR = Precambrian Research, GSAB = GSA Bulletin, GR = Gondwana Research, Tecto = Tectonics, JAES = Journal of Asian Earth Sciences, IGR = International Geology Review, BR = Basin Research, LithoSph = Lithosphere, Geol Mag, TectoPhy = Tectonophysics, SediGeol = Sedimentary Geology, ESR = Earth-Science Reviews, EPSL = Earth and Planetary Science Letters, JGS = Journal of the Geological Society, Geol J = Geological Journal, Int JES = International Journal of Earth Sciences, JSAmES = Journal of South American Earth Sciences, Acta PS = Acta Petrologica Sinica, GF = Geoscience Frontiers, Palaeo-3 = Palaeogeography, Palaeoclimatology, Palaeoecology, J Sedi R = Journal of Sedimentary Research.

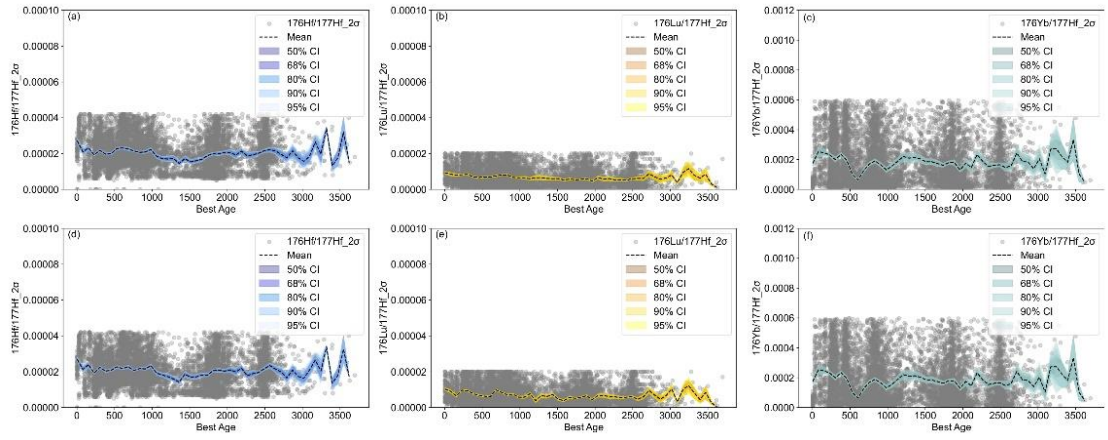

Figure S7: Temporal variations of isotopic uncertainties in Lu-Hf dataset. (a) $^{176}Hf/^{177}Hf$ with bootstrap resampling; (b) $^{176}Lu/^{177}Hf$ with bootstrap resampling; (c) $^{176}Yb/^{177}Hf$ with Monte-Carlo resampling; (d) $^{176}Hf/^{177}Hf$ with Monte-Carlo resampling; (e) $^{176}Lu/^{177}Hf$ with Monte-Carlo resampling; (f) $^{176}Yb/^{177}Hf$ with Monte-Carlo resampling.

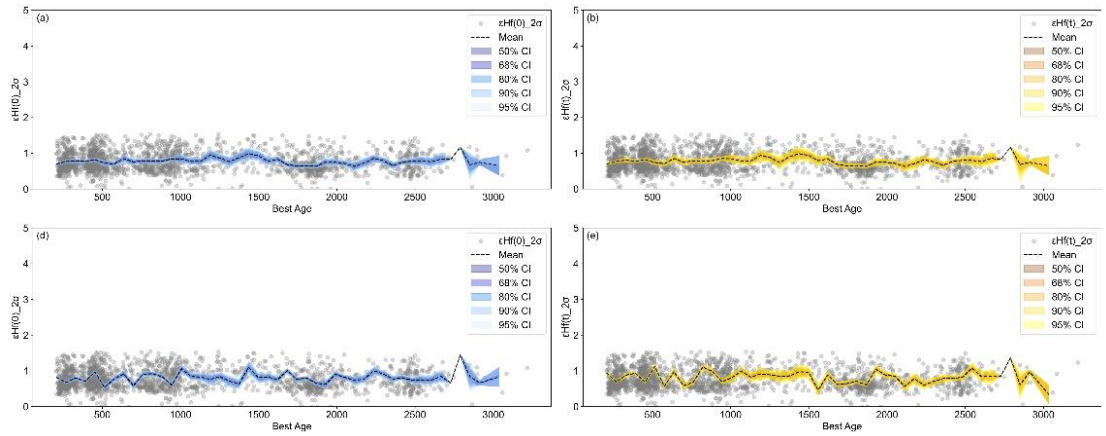

Figure S8: Temporal variations of εHf uncertainties in Lu-Hf dataset. (a) εHf(0) with bootstrap resampling; (b) εHf(t) with bootstrap resampling; (c) εHf(0) with Monte-Carlo resampling; (d) εHf(t) with Monte-Carlo resampling.

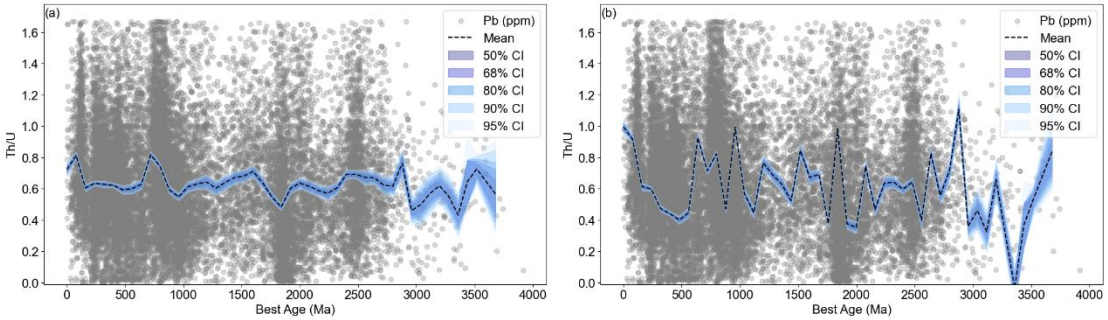

Figure S9: Temporal variations of Th/U in U-Pb dataset. (a) Th/U with bootstrap resampling; (b) Th/U with Monte-Carlo resampling.