The Journal of Open Source Software

A Python Tool for Predicting and Assessing Unconventional Rare-Earth and Critical Mineral Resources

Patrick Wingo [1,2¶], **Devin Justman**[1,2], **C. Gabriel Creason** [1,3], **Mackenzie Mark-Moser** [1,2], **Scott Montross** [1,2], and **Kelly Rose** [1]

1 National Energy Technology Laboratory (NETL), USA **2** NETL Support Contractor, USA **3** Oak Ridge Institute for Science and Education (ORISE), USA ¶ Corresponding author

DOI: 10.21105/joss.05500

Software

- Review
- Repository
- Archive

Editor: Jayaram Hariharan

Reviewers:

- @jeinsle
- @jameshgrn
- @FrancescoPerrone

Submitted: 29 March 2023
Published: 08 September 2023

Summary

Unconventional Rare-earth elements & Critical minerals (URC) (Creason, 2023) are crucial to a growing number of industries worldwide (Balaram, 2019). Due to their use in manufacturing, Critical Minerals (CM) are essential to economic and national security, yet have supply chains vulnerable to external disruptions (Yesenchak et al., 2022). *Unconventional* CM are sourced from geologic or byproduct hosts distinctly separate from the mechanisms which establish conventional CM deposits (Yesenchak et al., 2022). Unconventional sources for CM include *in situ* geologic deposits and byproducts of industrial extraction (Yesenchak et al., 2022).

The extraction and recovery of conventional CM is a complex process traditionally involving strip mining, which is both expensive and environmentally destructive (Balaram, 2019). Recent research has revealed that coaliferous sediments may act as unconventional CM sources containing Rare-Earth Elements (REE) in significant concentrations (Seredin & Dai, 2012). Determining the likelihood and location of REE resources in sedimentary basins is both complex and challenging. To address this, a new method of evaluating the potential occurrence of URC resources using a series of validated heuristics has been developed (Creason, 2023). While the entire process can be carried out manually using a collection of tools, a new, standalone software tool has been developed to streamline and expedite the process: NETL's `URC Resource Assessment Tool`.

Statement of need

The `URC Resource Assessment Tool` applies the data analysis methods outlined in Creason (2023), the tool's companion paper. This tool is a complete application written in Python and built on top of several open-source libraries (see the Support Libraries section). No other Python packages are known to contain the combination of geospatial information systems (GIS) and fuzzy logic support required to directly implement the method defined in Creason (2023). The intended users for this tool are geologists and geospatial scientists who are looking to better understand the mode and spatial distribution of potential URC resource occurrences in sedimentary basins.

There are several ways that the `URC Resource Assessment Tool` can be configured to run, but fundamentally the tool takes in a collection of spatial domains which fall under the Lithological, Structural, and Secondary Alteration categories defined by the Subsurface Trend Analysis (STA) method (Rose et al., 2020). These domains are combined, clipped to a researcher-defined boundary, and grided to cells of a research-specified dimension (see Figure 1 for overview of the process).

From this point, a Data Availability (DA) and / or a Data Supporting (DS) analysis can be undertaken by the tool (Figure 2); both analyses will operate on a vector-based spatial dataset describing the target formation, following the labelling scheme specified in the supplementary material in Creason (2023). These data are rasterized according to the grid specification of the aforementioned domains with each cell tagged with the appropriate set of indices. In the case of a DA analysis, each pixel in the rasterized data is evaluated by applying Equation (1) as described in Creason (2023), producing a DA score for each cell that is unique to each geologic resource type. For the DS analysis, the Spatial Implicit Multivariate Probability Assessment (SIMPA) method (Wingo et al., 2019) is applied using a series of predefined Fuzzy Logic statements, encapsulating the logic of Equations (2), (3), and (4) in Creason (2023).

The `URC Resource Assessment Tool` can be run either using the standalone GUI, or as a command-line tool. The former configuration is useful for a guided approach to executing the URC mineral-related analyses and previewing the results in the tool itself, whereas the latter is useful for integration of the tool into a workflow as part of a batch process. Regardless of how it is run, the results of the requested analyses are written to GeoTIFF files, which can be imported into most GIS analysis tools. Optionally, when run from the GUI the results of an analysis can be previewed within the tool itself (Figure 3).

Implementation Details

The `URC Resource Assessment Tool` relies on several existing open source libraries to perform its analyses. The Geospatial Data Abstraction Library (GDAL) is utilized for managing geospatial inputs and outputs, projection transformations, and converting vector layers into raster layers. Rasters are converted to two-dimensional NumPy arrays, during arithmetic processing to both reduce code complexity and potentially reduce time complexity through NumPy's hardware optimizations, such as Advanced Vector Extensions (AVX) utilization on Intel hardware (NumPy Developers, n.d.).

Data analyses pertaining to DA were carried out using Pandas. Raster information is converted into a Pandas DataFrame object, with each column representing a layer, and each row representing a pixel location. Sums are calculated according to the DA scoring algorithm outlined in Creason (2023), with the final results taken from the pandas Dataframe and converted into geospatial rasters.

The fuzzy logic statements driving the DS analysis are authored using the SIMPA tool, and then incorporated into the `URC Resource Assessment Tool` by using the embedded urclib.fuzzylogic package to convert the logic to Python. The collection of fuzzy logic statements are executed across all rasters on a per-pixel coordinate basis. This creates a Single Instruction, Multiple Data (SIMD) condition which is heavily parallelized using python's `multiprocessing` module, further reducing time complexity and noticeably reducing overall processing time.

For more information on how the fuzzy logic library works, see the SIMPA tool documentation (Wingo et al., 2019).

Support Libraries

In addition to several core Python libraries, The following 3rd-party libraries were used to create this tool:

- GDAL, *v3.1.4*: Used for general spatial data management and calculations (GDAL/OGR contributors, 2022).
- NumPy, *v1.23.2*: Handled vector math and general numeric array management (Harris et al., 2020).
- Pandas, *v1.2.5*: Utilized for statistical calculations for the *Data Available* (DA) scoring (The pandas development team, 2020).

- SIMPA, *v2.0.0*: Tool for processing spatially explicit raster data using fuzzy logic statements. Used in *Data Supporting* (DS) scoring (Wingo et al., 2019).
- PyQt, *v5.15.6*: Framework used to build the graphical user interface (GUI) for the tool; built on Qt (Qt Group, 2022; Riverbank Computing, 2022).
- PyOpenGL, *v3.1.6*: Wrapper for OpenGL API; used for map visualization (Fletcher, 2022; Khronos Group, 2017).
  - PyOpenGL-accelerate, *v3.1.6*: Optional library which can increase the performance of PyOpenGL (Fletcher, 2022).
- pyGLM, *v2.2.0*: Python port of the GLM library; used for graphic-specific mathematics (G-Truc, 2020; Zuzu_Typ, 2022).

Figures

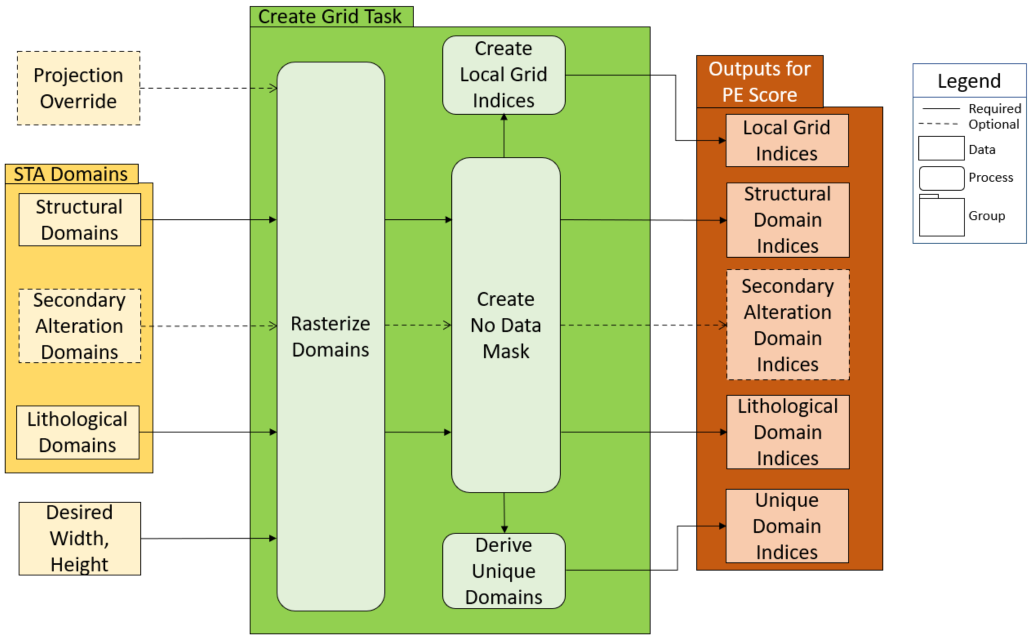

Figure 1: High level overview of the tool workflow when carrying out the *Create Grid* task. Domains created using the STA method (Rose et al., 2020) are rasterized using the provided width and height values for each pixel. If desired a geospatial projection can be specified to be assigned to the results by providing a projection or European Petroleum Survey Group (EPSG) code as a "projection override". This task produces a series of index rasters suitable as inputs to the *Potential Enrichment (PE) Score Task*.

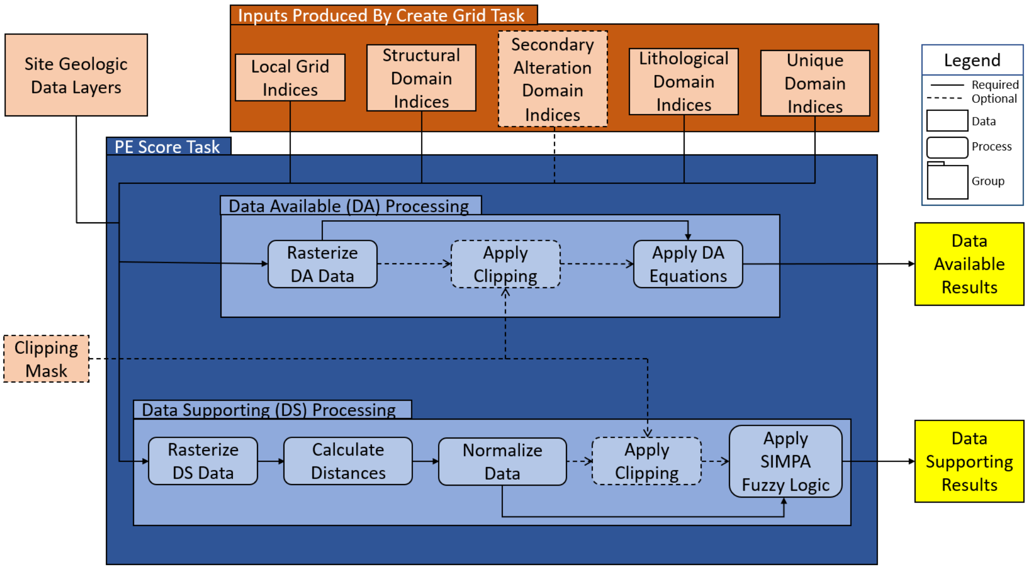

Figure 2: High level overview of the tool workflow when carrying out the *Potential Enrichment (PE) Score* task. Inputs for this task include index rasters generated from a previous execution of the *Create Grid Task*, a collection of site geologic data in the form of spatial vector layers, and an optional clipping mask. This information is used to produce analyses of both Data Available (DA) and Data Supporting (DS) layers provided as part of the site geologic data, providing insight into the likelihood of the presence of URC resources.

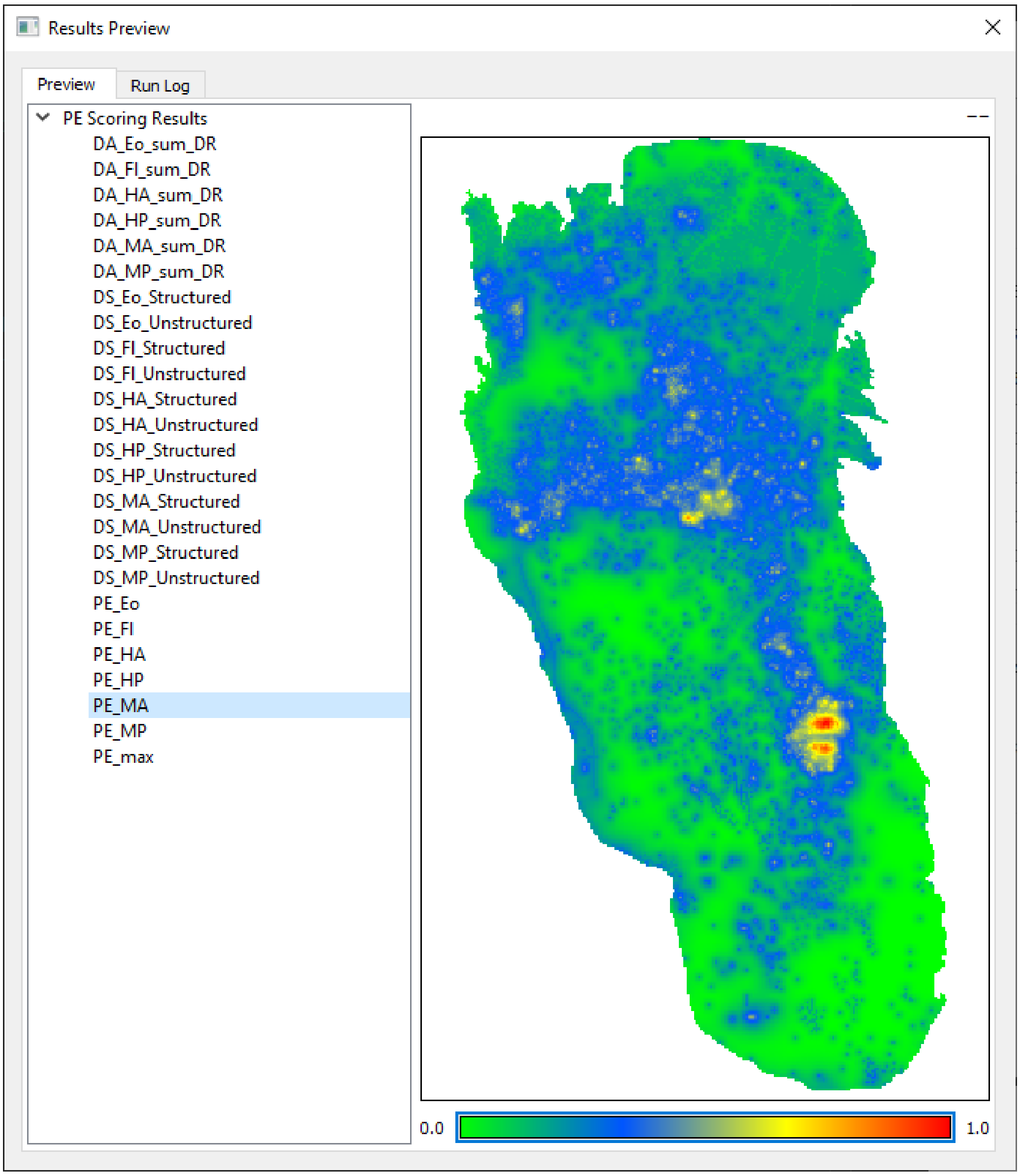

Figure 3: Example Result preview generated by URC Resource Assessment Method Tool; the selected output shows a map colored according to PE score for Meteoric Adsorption (MA). All spatially explicit outputs can be previewed with user-configurable color scales.

Acknowledgements & Disclaimer

Disclaimer: This project was funded by the U.S. Department of Energy, National Energy Technology Laboratory, in part, through a site support contract. Neither the United States Government nor any agency thereof, nor any of their employees, nor the support contractor, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the

United States Government or any agency thereof.

Acknowledgement: Parts of this technical effort were performed in support of the National Energy Technology Laboratory's (NETL) ongoing research under the Critical Minerals Field Work Proposal by NETL's Research and Innovation Center. The authors are grateful for literature synthesis provided by Jenny DiGuilio, Nicole Rocco, Roy Miller III, and Emily Cameron early in the development of the URC resource assessment method. Technical discussions with Davin Bagdonas, Leslie "Jingle" Ruppert, and Paige Morkner aided development and led to the advancement of the assessment method. Development and validation of the URC method benefited from geologic core and coal samples provided by the University of Wyoming, U.S. Geologic Survey, West Virginia Geological and Economic Survey, and Ramaco Carbon.

References

Balaram, V. (2019). Rare earth elements: A review of applications, occurrence, exploration, analysis, recycling, and environmental impact. *Geoscience Frontiers*, *10*(4), 1285–1303. https://doi.org/10.1016/j.gsf.2018.12.005

Creason, J., C. G. (2023). A geo-data science method for assessing unconventional rare-earth occurrences in sedimentary systems. *Natural Resources Research*. https://doi.org/10.1007/s11053-023-10163-x

Fletcher, M. C. (2022). PyOpenGL 3.x: The python OpenGL binding. In *Sourceforge Repository*. Sourceforge. https://pyopengl.sourceforge.net/

GDAL/OGR contributors. (2022). *GDAL/OGR geospatial data abstraction software library*. Open Source Geospatial Foundation. https://doi.org/10.5281/zenodo.5884351

G-Truc. (2020). OpenGL mathematics (GLM). In *Github Repository*. Github. https://glm.g-truc.net/0.9.9/index.html

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Khronos Group. (2017). *OpenGL: The industry's foundation for high performance graphics*. Khronos Group. https://www.opengl.org/

NumPy Developers. (n.d.). *CPU/SIMD optimizations - numpy v2.0dev0 manual*. NumPy Developers. https://numpy.org/devdocs/reference/simd/index.html

Qt Group. (2022). *Qt*. Qt Group. https://www.qt.io

Riverbank Computing. (2022). *PyQt*. Riverbank Computing. https://riverbankcomputing.com/software/pyqt/

Rose, K. K., Bauer, J. R., & Mark-Moser, M. (2020). A systematic, science-driven approach for predicting subsurface properties. *Interpretation*, *8*(1), T167–T181. https://doi.org/10.1190/INT-2019-0019.1

Seredin, V. V., & Dai, S. (2012). Coal deposits as potential alternative sources for lanthanides and yttrium. *International Journal of Coal Geology*, *94*, 67–93. https://doi.org/10.1016/j.coal.2011.11.001

The pandas development team. (2020). *Pandas-dev/pandas: pandas* (latest). Zenodo. https://doi.org/10.5281/zenodo.3509134

Wingo, P., Justman, D., Creason, G., Jones, K., Bauer, J., & Rose, K. (2019). SIMPA. In *EDX Submission*. EDX. https://doi.org/10.18141/1503876

Yesenchak, R., Justman, D., Bauer, S., Creason, C. G., Gordon, A., Montross, S. N., Sabbatino, M., & Rose, K. (2022). *Unlocking the potential of unconventional critical mineral resources story map*. https://doi.org/10.18141/1891489

Zuzu_Typ. (2022). PyGLM. In *Github Repository*. Github. https://github.com/Zuzu-Typ/PyGLM