The Journal of Open Source Software

gediDB: A toolbox for processing and providing Global Ecosystem Dynamics Investigation (GEDI) L2A-B and L4A-C data

Simon Besnard [1¶], **Felix Dombrowski** [2], **and Amelia Holcomb** [3]

1 GFZ Helmholtz Centre Potsdam, Potsdam, Germany **2** University of Potsdam, Potsdam, Germany **3** Department of Computer Science, University of Cambridge, Cambridge, UK ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Abstract

The Global Ecosystem Dynamics Investigation (GEDI) mission provides spaceborne LiDAR observations that are essential for characterising Earth's forest structure and carbon dynamics. However, GEDI datasets are distributed as complex HDF5 granules, which pose significant challenges for efficient, large-scale data processing and analysis. To overcome these hurdles, we developed `gediDB`, an open-source Python standardised Application Programming Interface (API) that streamlines both the processing and querying of GEDI Level 2A–B and Level 4A–C datasets. Built on the optimised multidimensional array database `TileDB`, `gediDB` enables operational-scale processing, rapid spatial and temporal queries, and reproducible LiDAR-based analyses of forest biomass, carbon stocks, and structural change.

Statement of Need

High-volume LiDAR datasets from the Global Ecosystem Dynamics Investigation (GEDI) mission (R. Dubayah et al., 2020) (Fig. 1) have become essential for quantifying forest dynamics, estimating biomass, and analysing carbon cycling. The open availability of GEDI's spaceborne LiDAR data has enabled forest structural analysis at near-global scales. However, practical use remains hindered by the complexity of raw HDF5 granules, the absence of scalable infrastructure for efficient access, and a lack of standardised tools for large-scale spatial and temporal subsetting.

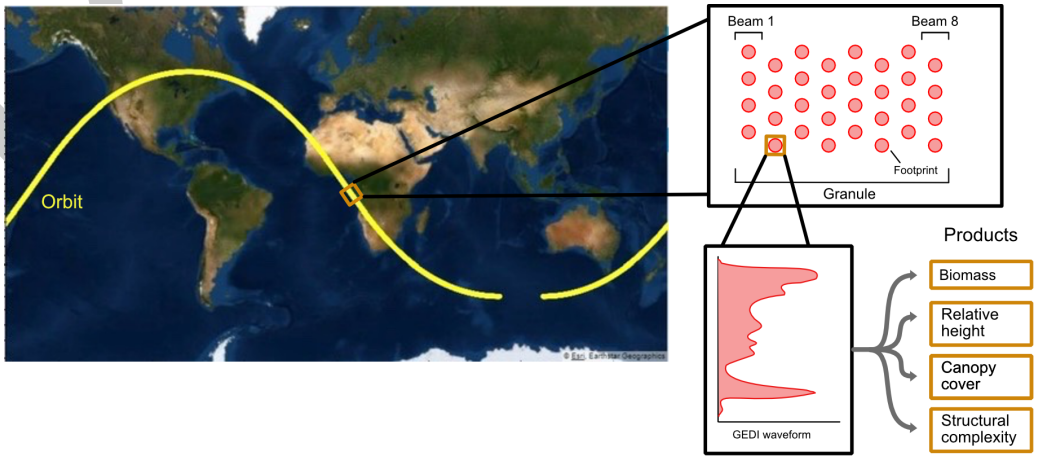

Fig. 1: A schematic representation of the GEDI data structure. Credits: Amelia Holcomb's PhD dissertation (Holcomb, 2025)

Several efforts in the broader NASA LiDAR community have tackled similar challenges. For example, SlideRule (Shean et al., 2023) provides a scalable, cloud-based framework for processing ICESat-2 photon data, enabling users to query and transform complex satellite LiDAR datasets into analysis-ready forms. This illustrates a common pattern: while raw LiDAR missions deliver highly relevant observations, the data formats and scales make direct scientific use difficult without specialised infrastructure. For GEDI, existing services such as NASA's GEDI Subsetter via the Multi-Mission Algorithm and Analysis Platform (MAAP) (Daniels et al., 2025) offer useful access for small to moderate-scale extractions, but they are not designed for operational-scale workflows or integration into reproducible pipelines.

`gediDB` addresses this gap by providing a robust, scalable Python-based API for unified access to GEDI Level 2A (R. Dubayah, Hofton, et al., 2021), 2B (R. Dubayah, Tang, et al., 2021), 4A (R. O. Dubayah et al., 2022), and 4C (De Conto et al., 2024) products. Built on the TileDB storage engine (TileDB, Inc., 2025), it enables fast querying of multidimensional arrays by spatial extent, temporal range, and variable selection. Seamless integration with geospatial libraries such as xarray (Hoyer & Hamman, 2017) and geopandas (Jordahl et al., 2020) ensures compatibility with reproducible workflows, from local machines to cloud and high-performance computing environments. By leveraging TileDB's advanced spatial indexing, `gediDB` simplifies and accelerates GEDI data access and analysis (see Fig. 2).

The increasing use of GEDI in global applications, such as canopy height mapping (Pauls et al., 2024), disturbance assessment (Holcomb et al., 2024), and forest degradation monitoring (Bourgoin et al., 2024), highlights the need for efficient, scalable tooling. By streamlining data access and enabling large-scale, reproducible workflows, `gediDB` fills this role for the GEDI community, complementing efforts like SlideRule in the ICESat-2 domain and supporting broader ecological monitoring and policy-relevant research.

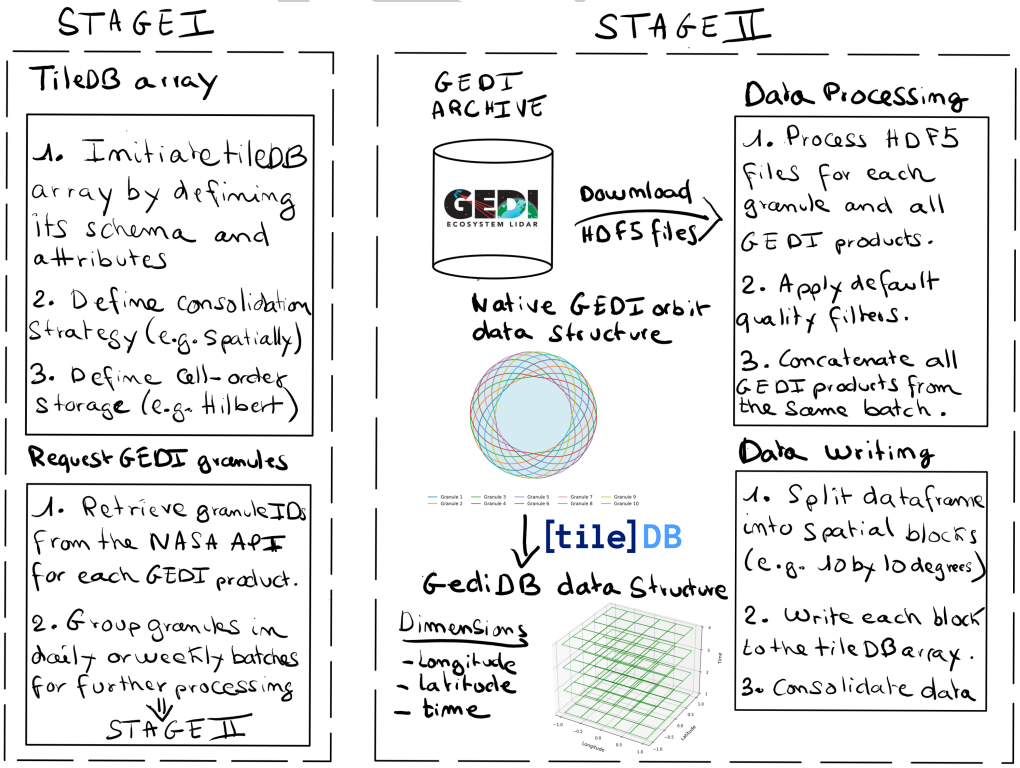

Fig. 2: A schematic representation of the gediDB data workflow.

The Journal of Open Source Software

Core functionalities

Extensive documentation and tutorials are available at https://gedidb.readthedocs.io, offering clear setup instructions, configuration guidance, and workflow examples. Users can access a globally processed GEDI dataset directly, avoiding the need for local downloads, as detailed in the database documentation.

Data processing framework

The `gediDB` package centres on two primary modules that streamline GEDI data ingestion and access:

- **`GEDIProcessor`**: Ingests raw GEDI granules and transforms them into structured TileDB arrays (Fig. 3). The process includes filtering, standardisation, and spatio-temporal chunking to ensure high-performance querying.
- **`GEDIProvider`**: Enables flexible access to GEDI data using spatial and temporal filters and variable selection. Output is compatible with Python libraries such as xarray and pandas (Reback et al., 2020).

Configurable and reproducible workflows

Custom configuration files define the TileDB schema and data retrieval parameters, supporting reproducibility and adaptability across diverse computing environments.

Robust data downloading

The API connects directly to NASA's Common Metadata Repository (CMR) and includes robust retry logic and error handling to ensure consistent, fault-tolerant data acquisition.

High-performance data storage

GEDI data are stored as sparse TileDB arrays optimised for fast spatial and temporal queries. The array structure accommodates large, multidimensional datasets efficiently (Fig. 3).

Parallel processing capabilities

`gediDB` supports parallelised downloading, processing, and storage using libraries such as `Dask` (Rocklin, 2015) and `concurrent.futures`, enabling high-throughput workflows on HPC systems.

Advanced querying functionality

`gediDB` offers flexible querying capabilities, including bounding-box, temporal range, and nearest-neighbour queries. Both scalar and profile-type variables are supported.

Rich metadata integration

Comprehensive metadata—covering provenance, units, variable descriptions, and versioning—is embedded within the TileDB arrays, ensuring transparency and reproducibility.

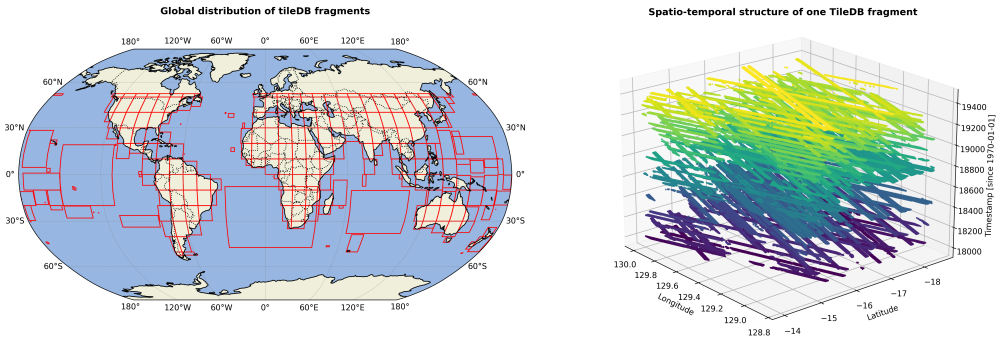

Fig. 3: Illustration of the global GEDI data storage schema using TileDB arrays.

Performance benchmarks

To evaluate the efficiency of `gediDB`, we benchmarked representative research scenarios and compared them to equivalent queries performed with NASA's GEDI Subsetter via the Multi-Mission Algorithm and Analysis Platform (MAAP). This comparison highlights not only absolute performance, but also practical trade-offs between the two approaches. The table below reports query times (in seconds) for varying spatial and temporal extents.

Scenario	Spatial extent	Time range	Variables queried	Query time (gediDB)	Query time (MAAP)
Local-scale query	1° × 1° bounding box	1 month	relative height metrics, canopy cover	1.8	*TBD*
Regional-scale query	10° × 10° bounding box	6 months	relative height metrics, biomass, plant area index	17.9	*TBD*
Continental-scale query	Amazon Basin	1 year	canopy cover, biomass	28.9	*TBD*

Benchmarks for `gediDB` were performed on a Linux server with dual Intel® Xeon® E5-2643 v4 CPUs (12 cores, 24 threads), 503 GB RAM, and NVMe SSD (240 GB) + HDD (16.4 TB) storage. All queries ran on NVMe-backed data to ensure high I/O throughput.

Example use cases

We used `gediDB` to analyse aboveground biomass and canopy cover dynamics across the Amazon Basin (Fig. 4). The workflow extracted variables including aboveground biomass, canopy cover, and relative height (RH) metrics across large spatial extents and multiple years. Data were aggregated within a 1°×1° hexagonal grid to support spatiotemporal analysis of forest structural change. The analysis pipeline was implemented entirely in Python using geopandas and `xarray`, making it fully reproducible from data extraction to visualisation.

The Journal of Open Source Software

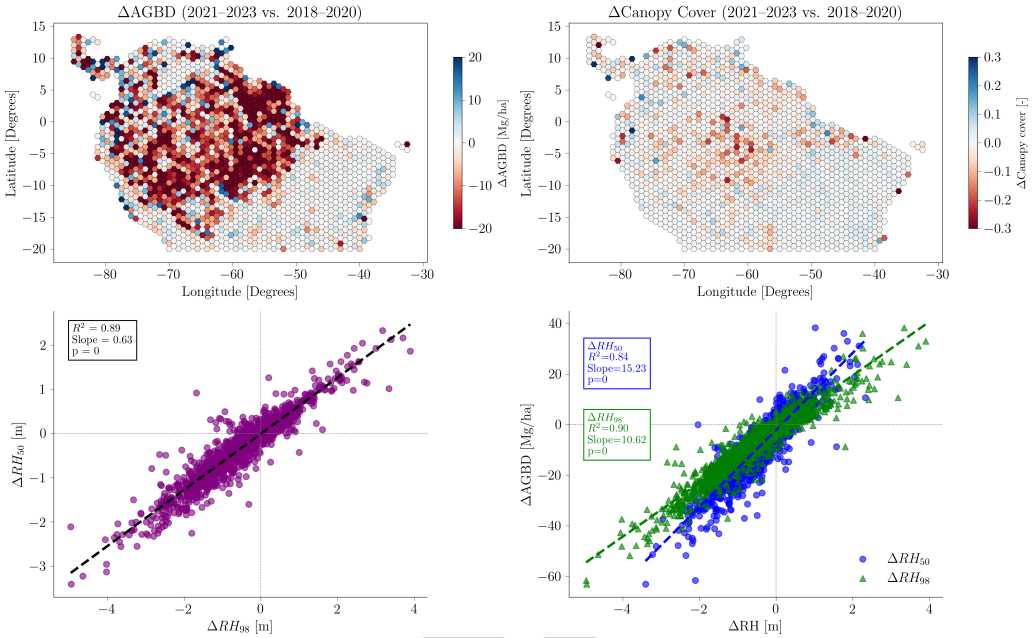

Fig. 4: Visualisation of changes in aboveground biomass density (AGBD) (top left panel) and canopy cover (top right panel) between 2018–2020 and 2021–2023, aggregated to a 1°×1° hexagonal grid over the Amazon Basin. The bottom left panel shows the relationship between changes in ΔRH50 and ΔRH98, with each point representing a hexagon. The bottom right panel shows the relationship between changes in canopy height metrics (ΔRH50 and ΔRH98) and ΔAGBD, with each point representing a hexagon. This highlights how vertical canopy structure dynamics relate to biomass change across the region.

A key advantage of `gediDB` is that large-scale extractions can be performed directly within Python workflows, eliminating the need for manual downloads or interactive tools such as MAAP. For example, the following snippet retrieves biomass, canopy cover, and RH metrics for the Amazon Basin:

```
import geopandas as gpd
import gedidb as gdb

# Instantiate provider with S3 backend
provider = gdb.GEDIProvider(
    storage_type="s3",
    s3_bucket="dog.gedidb.gedi-l2-l4-v002",
    url="https://s3.gfz-potsdam.de"
)

# Load region of interest (Amazon Basin)
roi = gpd.read_file("amazon_basin.geojson")

# Query GEDI data as xarray dataset
ds = provider.get_data(
    variables=["agbd", "cover", "rh_98", "rh_50"],
    query_type="bounding_box",
    geometry=roi,
    start_time="2018-01-01",
    end_time="2024-01-01",
    return_type="xarray"
)
```

Future development

Planned future developments for `gediDB` are designed to improve usability and extend the package's scope for both researchers and operational users:

- **Compatibility with upcoming GEDI product releases**: ensures long-term sustainability of the toolbox as new mission data become available, avoiding version lock-in for users building workflows on `gediDB`.
- **Improved performance and flexibility in querying profile variables**: will make it easier for users to analyse canopy structure profiles (e.g., RH metrics) at scale, which are currently among the most data-intensive GEDI products.
- **Support for direct HDF5 access from AWS S3**: will enable `gediDB` to operate directly on cloud-hosted GEDI granules (e.g., on NASA's MAAP infrastructure), avoiding the need for local downloads and reducing storage overhead. This work corresponds to Issue #15 ("Allow direct h5 read w/o download"), which aims to let the `GEDIProcessor` access HDF5 files in an S3 bucket without intermediate steps.
- **Expanded documentation and tutorials**: will benefit new users by lowering the entry barrier, providing clear end-to-end examples, and connecting scientific use cases to code snippets.
- **Strengthened testing for reliability and maintainability**: supports developers and long-term users by ensuring that changes do not break existing workflows, and by increasing trust in the reproducibility of analyses built on `gediDB`.

Development progress and discussion of these features are tracked openly through the project's GitHub issues and roadmap.

Conclusion

`gediDB` enhances the usability of GEDI LiDAR datasets by addressing challenges of data complexity, scalability, and reproducibility. Built on TileDB, it enables efficient data management, fast querying, and integration into geospatial workflows—facilitating large-scale analysis of forest dynamics and carbon cycling. Its open-source, community-driven design supports ongoing progress in remote sensing and environmental science.

Acknowledgements

The development of `gediDB` was supported by the European Union through the FORWARDS and NextGenCarbon projects. We also acknowledge funding for 3D-ABC by the Helmholtz Foundation Model Initiative, supported by the Helmholtz Association. We would also like to thank the R2D2 Workshop (March 2024, Potsdam) for providing the opportunity to meet and discuss GEDI data processing. We recognise using OpenAI's ChatGPT and Grammarly AI tools to enhance the manuscript's sentence structure, conciseness, and grammatical accuracy.

References

Bourgoin, C., Ceccherini, G., Girardello, M., Vancutsem, C., Avitabile, V., Beck, P., Beuchle, R., Blanc, L., Duveiller, G., Migliavacca, M., & others. (2024). Human degradation of tropical moist forests is greater than previously estimated. *Nature*, *631*(8021), 570–576. https://doi.org/10.1038/s41586-024-07629-0

Daniels, C., French, J., Adhikari, S., Bhusal, A., Mandel, A. I., & Kirkland, S. (2025). *MAAP-project/gedi-subsetter: 0.10.0* (Version 0.10.0). Zenodo. https://doi.org/10.5281/zenodo.15122227

De Conto, T., Armston, J., & Dubayah, R. O. (2024). *GEDI L4C Footprint Level Waveform Structural Complexity Index, Version 2*. ORNL DAAC, Oak Ridge, Tennessee, USA. https://doi.org/10.3334/ORNLDAAC/2338

Dubayah, R. O., Armston, J., Kellner, J. R., Duncanson, L., Healey, S. P., Patterson, P. L., Hancock, S., Tang, H., Bruening, J. M., Hofton, M. A., Blair, J. B., & Luthcke, S. B. (2022). *GEDI L4A Footprint Level Aboveground Biomass Density, Version 2.1*. ORNL DAAC, Oak Ridge, Tennessee, USA. https://doi.org/10.3334/ORNLDAAC/2056

Dubayah, R., Blair, J. B., Goetz, S., Fatoyinbo, L., Hansen, M., Healey, S., Hofton, M., Hurtt, G., Kellner, J., Luthcke, S., Armston, J., Tang, H., Duncanson, L., Hancock, S., Jantz, P., Marselis, S., Patterson, P. L., Qi, W., & Silva, C. (2020). The global ecosystem dynamics investigation: High-resolution laser ranging of the earth's forests and topography. *Science of Remote Sensing*, *1*, 100002. https://doi.org/10.1016/j.srs.2020.100002

Dubayah, R., Hofton, M., Blair, J., Armston, J., Tang, H., & Luthcke, S. (2021). *GEDI L2A Elevation and Height Metrics Data Global Footprint Level V002*. NASA EOSDIS Land Processes Distributed Active Archive Center. https://doi.org/10.5067/GEDI/GEDI02_A.002

Dubayah, R., Tang, H., Armston, J., Luthcke, S., Hofton, M., & Blair, J. (2021). *GEDI L2B Canopy Cover and Vertical Profile Metrics Data Global Footprint Level V002*. NASA EOSDIS Land Processes Distributed Active Archive Center. https://doi.org/10.5067/GEDI/GEDI02_B.002

Holcomb, A. (2025). *Measuring tropical forest disturbance and regrowth with spaceborne lidar*. https://www.repository.cam.ac.uk/handle/1810/389269

Holcomb, A., Burns, P., Keshav, S., & Coomes, D. A. (2024). Repeat GEDI footprints measure the effects of tropical forest disturbances. *Remote Sensing of Environment*, *308*, 114174. https://doi.org/10.1016/j.rse.2024.114174

Hoyer, S., & Hamman, J. J. (2017). Xarray: N-d labeled arrays and datasets in python. *Journal of Open Research Software*, *5*(1), 10. https://doi.org/10.5334/jors.148

Jordahl, K., Bossche, J. V. den, Fleischmann, M., Wasserman, J., McBride, J., Gerard, J., Tratner, J., Perry, M., Badaracco, A. G., Farmer, C., Hjelle, G. A., Snow, A. D., Cochran, M., Gillies, S., Culbertson, L., Bartos, M., Eubank, N., maxalbert, Bilogur, A., … Leblanc, F. (2020). *Geopandas/geopandas: v0.8.1* (Version v0.8.1). Zenodo. https://doi.org/10.5281/zenodo.3946761

Pauls, J., Zimmer, M., Kelly, U. M., Schwartz, M., Saatchi, S., Ciais, P., Pokutta, S., Brandt, M., & Gieseke, F. (2024). *Estimating canopy height at scale*. https://doi.org/10.48550/arXiv.2406.01076

Reback, J., McKinney, W., jbrockmendel, Van den Bossche, J., Augspurger, T., Cloud, P., Hawkins, S., Gfyoung, Sinhrks, Klein, A., Roeschke, M., & Tratner, W. (2020). Pandas-dev/pandas: pandas. *Zenodo*. https://doi.org/10.5281/zenodo.3509134

Rocklin, M. (2015). Dask: Parallel computation with blocked algorithms and task scheduling. *Proceedings of the 14th Python in Science Conference*, 130–136. https://doi.org/10.25080/majora-7b98e3ed-013

Shean, D., Swinski, J. p., Smith, B., Sutterley, T., Henderson, S., Ugarte, C., Lidwa, E., & Neumann, T. (2023). SlideRule: Enabling rapid, scalable, open science for the NASA ICESat-2 mission and beyond. *Journal of Open Source Software*, *8*(81), 4982. https://doi.org/10.21105/joss.04982

TileDB, Inc. (2025). *Tiledb: Modern database engine for complex data based on multi-dimensional arrays*. https://github.com/TileDB-Inc/TileDB-Py