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Published: unpublished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 standardized 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 a key resource for quantifying forest dynamics, estimating biomass, and analysing carbon cycling. The open availability of GEDI's spaceborne LiDAR data has created unprecedented opportunities to extend forest structural analyses from local or regional case studies to near-global scales. However, despite the richness of information contained in GEDI datasets, their practical usability remains challenging due to the complexity of raw HDF5 granules, a lack of scalable infrastructure for efficient data retrieval, and insufficient standardized tools for large-scale spatial and temporal subsetting.

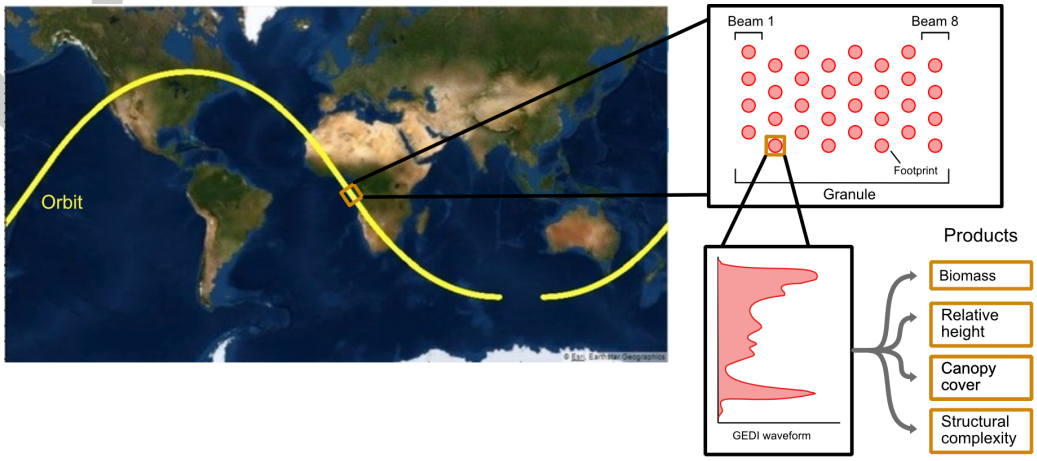

Fig. 1: A schematic representation of the GEDI data structure. Credits: Amelia Holcomb's

PhD dissertation

Existing software tools for GEDI data analysis, such as the GEDI Subsetter provided by NASA's Multi-Mission Algorithm and Analysis Platform (MAAP) [GEDI Subsetter; Daniels et al. (2025)], primarily address small to moderate-scale data extraction scenarios. While suitable for interactive or limited spatial extents, these tools often struggle to efficiently support large-scale workflows, leading to computational bottlenecks and decreased efficiency when applied to extensive spatial and temporal analyses.

`gediDB` addresses these limitations by offering a robust and scalable framework that unifies access to GEDI Level 2A (R. Dubayah, Hofton, et al., 2021), Level 2B (R. Dubayah, Tang, et al., 2021), Level 4A (R. O. Dubayah et al., 2022) and Level 4C (De Conto et al., 2024) data via an Python (Van Rossum & Drake, 2009) standardized Application Programming Interface (API). Built on the TileDB storage engine, `gediDB` supports rapid querying of multidimensional arrays, allowing users to efficiently extract large data subsets by spatial extent, temporal range, and variable selection. It integrates seamlessly with Python's geospatial data ecosystem, including libraries such as `xarray` (Hoyer & Hamman, 2017) and geopandas (Jordahl et al., 2020), and integrates into reproducible workflows that can scale to high-performance computing environments and cloud platforms. By leveraging TileDB's advanced spatial indexing, `gediDB` substantially simplifies the processing and querying of GEDI data (see Fig. 2).

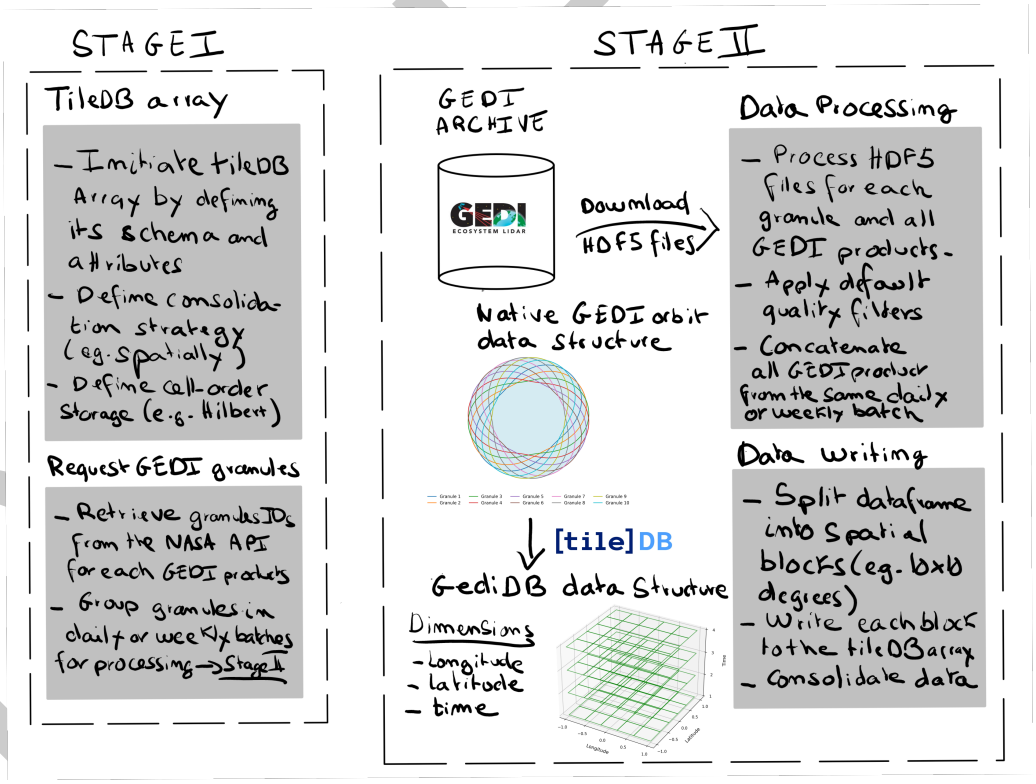

Fig. 2: A schematic representation of the gediDB data workflow.

Core functionalities

Extensive documentation and user tutorials for `gediDB` are available at https://gedidb.readthedocs.io. These resources provide comprehensive setup instructions, configuration guidance, and workflow examples. Users have immediate access to a globally processed GEDI dataset, eliminating the need for local downloads, as detailed in the database documentation.

Data processing framework

The `gediDB` package is structured around two core modules designed to streamline GEDI data processing and retrieval:

- **`GEDIProcessor`**: Systematically ingests raw GEDI granules and transforms them into structured TileDB arrays (Fig. 3). Key steps include data filtering, standardisation, and efficient spatio-temporal chunking to ensure optimal query performance.
- **`GEDIProvider`**: Enables rapid, flexible access to GEDI data using spatial bounding boxes, temporal filters, and user-selected variables. Results are provided in formats fully compatible with Python's geospatial libraries such as `xarray` and `pandas` (Reback et al., 2020).

Configurable and reproducible workflows

`gediDB` uses customisable configuration files to define TileDB schemas and data retrieval parameters. This facilitates reproducibility and adaptability across diverse research scenarios and computing environments.

Robust data downloading

The API interfaces directly with NASA's Common Metadata Repository (CMR) to facilitate reliable data acquisition. It incorporates comprehensive retry logic and robust error handling to mitigate issues related to network interruptions or data inconsistencies.

High-performance data storage

GEDI data is efficiently stored using structured TileDB sparse arrays optimised for rapid spatial and temporal queries. The array structure is specifically designed to handle large-scale, multi-dimensional data seamlessly (Fig. 3).

Parallel processing capabilities

Parallelised operations are fully supported, including data downloading, processing, and storage. `gediDB` leverages libraries such as `Dask` (Rocklin, 2015) and Python's built-in `concurrent.futures` to maximise performance on large-scale workflows and high-performance computing infrastructures.

Advanced querying functionality

`gediDB` offers sophisticated querying methods, enabling spatial and temporal subsetting through bounding-box, time-range, and nearest-neighbour queries. The system efficiently handles both scalar and profile-type variables.

Comprehensive metadata management

Rich metadata is systematically captured and managed, including data provenance, variable units, descriptions, and product version details. Metadata is embedded directly within the TileDB structure to facilitate clear data documentation and reproducibility.

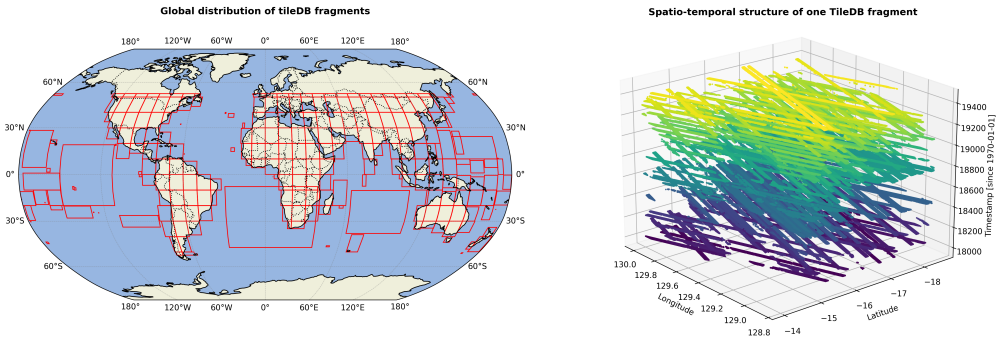

Fig. 3: Illustration of the global GEDI data storage schema using TileDB arrays.

Performance benchmarks

The efficiency of `gediDB` was evaluated under realistic research scenarios. The table below summarises query times across different spatial and temporal extents:

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

Benchmarks were conducted on a Linux server equipped with dual Intel® Xeon® E5-2643 v4 CPUs (12 physical cores, 24 threads total), 503 GB RAM, and a combination of NVMe SSD (240 GB) and HDD storage (16.4 TB total). Queries were executed from NVMe-backed storage to ensure high I/O performance. Compared to workflows based on direct HDF5 access, `gediDB` provides a significant speedup and streamlined user experience.

Example use cases

An illustrative use case involved the analysis of aboveground biomass and canopy cover dynamics across the Amazon Basin (Fig. 4). Leveraging `gediDB`, variables representing aboveground biomass, canopy cover, and vertical canopy structure (i.e., relative height (RH) metrics) were efficiently extracted over large spatial extents and multiple years. The data were aggregated within a 1°×1° hexagonal grid framework, enabling spatiotemporal analysis of forest structure change. Integration with Python's geospatial libraries, such as `geopandas` and `xarray`, allowed for a fully reproducible workflow from data extraction to visualization. To explore structural drivers of biomass change, a scatter plot analysis compared changes in upper and lower canopy height metrics (ΔRH98 and ΔRH50) with ΔAGBD. The greater slope observed between changes in median canopy height (ΔRH50) and aboveground biomass density (ΔAGBD), compared to changes in upper canopy height (ΔRH98) (lower right panel), indicates that biomass dynamics are more closely linked to widespread structural adjustments across the lower and mid-canopy layers rather than to changes restricted to the tallest forest emergents.

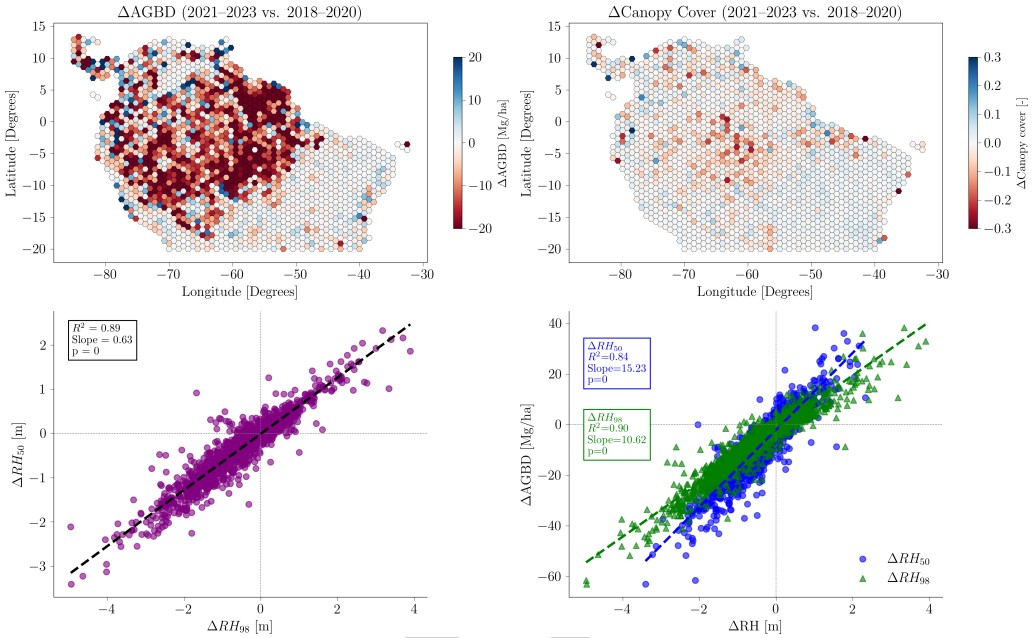

Fig. 4: Visualisation of changes in aboveground biomass density (AGBD) (top left panel) and canopy cover (top right panel) between 2018–2020 and 2021–2023, aggregated to a 1°×1° hexagonal grid over the Amazon Basin. The bottom left panel shows the relationship between changes in ΔRH50 and ΔRH98, with 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.

Beyond regional change assessments, `gediDB` supports advanced analyses such as biome-level comparisons of forest structural profiles, precise retrieval of GEDI data near field plots for calibration and validation, and the production of spatially gridded datasets at diverse resolutions. Recent studies have leveraged GEDI data to map canopy height across global forested ecosystems (Pauls et al., 2024), to assess forest disturbances (Holcomb et al., 2024), and to characterise forest degradation dynamics (Bourgoin et al., 2024). These applications demonstrate the potential of GEDI data to inform ecological monitoring and policy development. By streamlining data access, subsetting, and integration into scalable workflows, `gediDB` can significantly enhance the efficiency and reproducibility of such large-scale analyses—supporting efforts like global canopy height mapping, disturbance detection, and forest degradation monitoring.

Community impact and future development

`gediDB` fosters an open and collaborative research environmentby actively encouraging community-driven development through its GitHub repository. Its open-source nature promotes transparency, reproducibility, and long-term accessibility, benefiting a wide range of scientific applications in LiDAR research and environmental analysis.

Planned future developments for `gediDB` include:

- Maintaining compatibility with new GEDI data releases and product updates
- Enhancing performance and flexibility in querying profile variables
- Supporting direct reading of native HDF5 files from AWS S3 buckets
- Expanding tutorials and documentation to reach a broader user base
- Improving the testing framework to ensure greater reliability and maintainability

Feedback, feature requests, and code contributions from users and developers are warmly

welcomed. Through sustained community engagement, gediDB aims to continually evolve as a robust and adaptable tool for forest remote sensing and ecosystem monitoring.

Conclusion

gediDB substantially significantly improves the practical usability of GEDI LiDAR datasets by addressing key challenges related to data complexity, scalability, and reproducibility. Leveraging TileDB's optimised multidimensional array storage, it enables efficient data management, fast querying, and seamless integration into diverse geospatial analysis workflows. This empowers the systematic exploration of forest dynamics and carbon cycling at unprecedented spatial and temporal scales. Through its open-source, community-driven design, gediDB fosters collaborative progress in remote sensing, ecology, and environmental science, supporting the evolving needs of the Earth observation community.

Acknowledgements

The development of gediDB was supported by the European Union through the FORWARDS and OpenEarthMonitor projects. We would also like to acknowledge the R2D2 Workshop (March 2024, Potsdam) for providing the opportunity to meet and discuss GEDI data processing.

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.

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., & others. (2020). The global ecosystem dynamics investigation: High-resolution laser ranging of the earth's forests and topography. *Science of Remote Sensing*, *1*, 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., 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/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://arxiv.org/abs/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

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace. ISBN: 1441412697