The Journal of Open Source Software

PyForestScan: A Python library for calculating forest structural metrics from lidar point cloud data

Joseph Emile Honour Percival[1,2¶] **and Benjamin Palsa Leamon**[3]

1 Department of Geography & Environmental Science, University of Hawaiʻi at Hilo, Hilo, HI 96720, USA **2** Spatial Data Analysis & Visualization Laboratory, University of Hawaiʻi at Hilo, Hilo, HI 96720, USA **3** Independent Researcher, Japan ¶ Corresponding author

DOI: 10.21105/joss.07314

Software

- Review
- Repository
- Archive

Editor: Mengqi Zhao

Reviewers:

- @npucino
- @giswqs

Submitted: 21 September 2024
Published: 21 January 2025

Summary

`PyForestScan` is an open-source Python library designed for calculating forest structural metrics from Light Detection and Ranging (lidar) point cloud data at scale. The software calculates key ecological metrics such as foliage height diversity (FHD), plant area density (PAD), canopy height, plant area index (PAI), and digital terrain models (DTMs), efficiently handles large-scale lidar datasets, and supports input formats including the Entwine Point Tile (EPT) format (Manning, 2024), .las, .laz, and .copc files. In addition to metrics computation, the library supports the generation of GeoTIFF outputs and integrates with geospatial libraries like the Point Cloud Data Abstraction Library (`PDAL`) (Butler et al., 2021, 2024), making it a valuable tool for forest monitoring, carbon accounting, and ecological research.

Statement of Need

Remote sensing data, particularly point cloud data from airborne lidar sensors, are becoming increasingly accessible, offering a detailed understanding of forest ecosystems at fine spatial resolutions over large areas. This data is useful for calculating forest structural metrics such as canopy height, canopy cover, PAI, PAD, FHD, as well as DTMs, which are essential for forest management, biodiversity conservation, and carbon accounting (Drake et al., 2002; Guerra-Hernández et al., 2024; McElhinny et al., 2005; Pascual et al., 2020, 2021; Pascual & Guerra-Hernández, 2023).

Despite Python's prominence as a powerful language for geospatial and ecological analysis, there is a notable lack of dedicated, open-source tools within the Python ecosystem specifically designed for calculating comprehensive forest structural metrics from airborne lidar point-cloud data. This gap is significant given Python's extensive libraries for data science and its increasingly important role in ecology and deep learning (Borowiec et al., 2022). Existing open-source solutions that offer some of these metrics are primarily available in the R programming language. For instance, `lidR` (Roussel et al., 2020; Roussel & Auty, 2024) provides functions for point cloud manipulation, metric computation, and visualization but lacks native calculations for FHD and PAI. Another tool, `leafR` (Almeida et al., 2021), calculates FHD, leaf area index (LAI), and leaf area density (LAD) - both of which are very similar to PAI and PAD - but is limited in processing large datasets due to the absence of tiling functionality. Moreover, the importance of scale in lidar-based analyses of forest structure is well-documented (Atkins et al., 2023), and `leafR` does not allow users to modify voxel depth, which can be important for accurate estimation of structural metrics across different forest types and scales. Similarly, `canopyLazR` (Kamoske et al., 2019) provides tools to calculate LAD and LAI from point cloud lidar data but only allows the calculation of these metrics and also lacks support for tiling mechanisms, limiting its applicability to large datasets. Proprietary solutions like LAStools (LAStools, 2022), FUSION (McGaughey, 2022), and Global Mapper (Blue Marble Geographics,

2024) offer tools to calculate some of these metrics -mostly canopy height- but may not provide the flexibility required for diverse ecological contexts and are often inaccessible due to licensing costs. This lack of a comprehensive, scalable Python-based solution makes it challenging for researchers, ecologists, and forest managers to integrate point-cloud-based analysis into their Python workflows efficiently. This is particularly problematic when working with large datasets or when integrating analyses with other Python-based tools, such as those used for processing space-based waveform lidar data from the Global Ecosystem Dynamics Investigation (GEDI) mission (Dubayah et al., 2020; Tang & Armston, 2019), which also provides data on PAI, plant area volume density (PAVD), and FHD.

In addition to the lack of Python-based software for calculating forest structural metrics like PAI, PAD, and FHD, working with large-scale point clouds remains a challenge due to complexities inherent in the size of the data. Lidar datasets can vary in point densities—from about 2-3 points per square meter in airborne surveys covering vast landscapes to upwards of 2,000 points per square meter in high-resolution drone-based surveys, potentially resulting in terabytes of data. To manage these large volumes, datasets are typically divided into fixed-size tiles, which must be individually loaded into memory for analysis. This approach can introduce inflexibility because analyses may need to conform strictly to tile boundaries, potentially causing boundary effects when calculating metrics that span across tiles. While tools like `lidR` can handle tiling and mitigate boundary effects natively, they do not fully leverage the advanced spatial indexing provided by formats like EPT (Manning, 2024) and Cloud Optimized Point Cloud (COPC) (Butler & Contributors, 2021). Additionally, fixed tile sizes may not align with varying memory constraints or specific workflow needs, limiting the ability to adjust tile sizes dynamically based on data density and processing requirements. For example, extracting point clouds over specific polygons within tiles, or performing exploratory data analysis over a large region consisting of several tiles can be overly time-consuming as it often requires reading all data into memory.

`PyForestScan` was developed to fill this gap by providing an open-source, Python-based solution to calculate forest structural metrics that can handle large-scale point-cloud data while remaining accessible and efficient. By leveraging IO capabilities of `PDAL`, it handles large-scale analyses by allowing users to work with more efficient point-cloud data structure, such as spatially indexed hierarchical octree formats like EPT or COPC. In addition to lidar-based point clouds, it can also process point clouds derived from structure-from-motion (SfM) in open-canopy forests, provided the SfM data includes a sufficient density of points to capture the full vertical profile of the forest. `PyForestScan` supports commonly used formats such as .las, .laz, as well as more efficient formats such as COPC and EPT, and integrates with well-established geospatial frameworks for point clouds like `PDAL` (Butler et al., 2021, 2024). The more mathematically intensive calculations of PAD, PAI, and FHD are calculated following established methods by Kamoske et al. (2019) and Hurlbert (1971), and are given by equations (1) - (3). `PyForestScan` provides native tiling mechanisms to calculate metrics across large landscapes, IO support across multiple formats, point cloud processing tools to filter points and create ground surfaces, as well as simple visualization functions for core metrics. `PyForestScan` brings this functionality to Python, while also introducing capabilities not found in any single existing open-source software. By focusing on forest structural metrics, `PyForestScan` provides an essential tool for the growing need to analyze forest structure at scale in the context of environmental monitoring, conservation, and climate-related research.

$$PAD_{i-1,i} = \ln\left(\frac{S_e}{S_t}\right)\frac{1}{k\Delta z} \tag{1}$$

$$PAI = \sum_{i=1}^{n} PAD_{i-1,i} \tag{2}$$

$$FHD = -\sum_{i=1}^{n} p_i \ln(p_i) \tag{3}$$

In equation (1), $PAD_{i-1,i}$ represents the PAD between two adjacent voxels in the canopy, denoted by the indices $i-1$ and i. It quantifies the density of plant material within this vertical slice of the forest. Here, $\ln\left(\frac{S_e}{S_t}\right)$ calculates the natural logarithm of the ratio between the number of lidar pulses entering a voxel (S_e) and the number of pulses exiting the voxel (S_t), and $\frac{1}{k\Delta z}$ is the inverse of the extinction coefficient (k) as derived from the Beer-Lambert Law, multiplied by the height of each voxel (Δz).

Equation (2) represents PAI as the vertical summation of PAD across all layers i through n, as derived in equation (1).

In equation (3), FHD is calculated as the Shannon entropy of the vertical distribution of plant material across all layers of the canopy. p_i is the proportion of total plant material in each voxel i, relative to the entire vertical column, with n representing the number of vertical layers.

Usage

To facilitate usage of the software, we have included Jupyter notebooks in the GitHub repository detailing how to get started using `PyForestScan` as well as how to calculate forest metrics. The Jupyter notebooks include an example data set of a point cloud with a nominal pulse spacing of 0.35 meters and was captured over a dry forest environment. This example dataset is a one-square-kilometer tile derived from a 2018-2020 aerial lidar survey of the Big Island of Hawaii (Office for Coastal Management, 2024). The data has been preprocessed to classify ground and vegetation points (Guerra-Hernandez & Pascual, 2024). More details are available in the documentation.

Contributions

JEHP developed the concept with input from BPL; JEHP wrote the initial versions of the software and automatic tests with contributions from BPL; BPL and JEHP wrote the software documentation and created Jupyter notebooks for example usage; and both authors wrote the manuscript.

Acknowledgements

We would like to express our gratitude to Juan Guerra-Hernandez and Adrian Pascual for providing a noise-free classified point cloud dataset (Guerra-Hernandez & Pascual, 2024), which was instrumental in testing and validating the PyForestScan library. We would also like to thank Ryan Perroy for his guidance, feedback, and help in revising this manuscript. This work was enabled in part by funding from the National Science Foundation award: 2149133. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Almeida, D. R. A. de, Stark, S. C., Silva, C. A., Hamamura, C., & Valbuena, R. (2021). *leafR: Calculates the Leaf Area Index (LAD) and Other Related Functions* [Manual]. https://doi.org/10.32614/CRAN.package.leafR

Atkins, J. W., Costanza, J., Dahlin, K. M., Dannenberg, M. P., Elmore, A. J., Fitzpatrick, M. C., Hakkenberg, C. R., Hardiman, B. S., Kamoske, A., LaRue, E. A., Silva, C. A., Stovall, A. E. L., & Tielens, E. K. (2023). Scale dependency of lidar-derived forest structural diversity. *Methods in Ecology and Evolution*, *14*(2), 708–723. https://doi.org/10.1111/2041-210X.14040

Blue Marble Geographics. (2024). *Global mapper: GIS and mapping software*. https://www.bluemarblegeo.com/global-mapper/

Borowiec, M. L., Dikow, R. B., Frandsen, P. B., McKeeken, A., Valentini, G., & White, A. E. (2022). Deep learning as a tool for ecology and evolution. *Methods in Ecology and Evolution*, *13*(8), 1640–1660. https://doi.org/10.1111/2041-210X.13901

Butler, H., Bell, A., Gerlek, M. P., chambbj, Gadomski, P., Manning, C., Łoskot, M., Couwenberg, B., Barker, N., Ramsey, P., Dark, J., Mann, K., Chaulet, N., Rouault, E., Villemin, G., Foster, C., Moore, O., Rosen, M., Lewis, S., … Brookes, D. (2024). *PDAL/PDAL: 2.8.1* (Version 2.8.1). Zenodo. https://doi.org/10.5281/zenodo.13993879

Butler, H., Chambers, B., Hartzell, P., & Glennie, C. (2021). PDAL: An open source library for the processing and analysis of point clouds. *Computers & Geosciences*, *148*, 104680. https://doi.org/10.1016/j.cageo.2020.104680

Butler, H., & Contributors. (2021). *Cloud optimized point cloud (COPC)*. https://copc.io/

Drake, J. B., Dubayah, R. O., Clark, D. B., Knox, R. G., Blair, J. B., Hofton, M. A., Chazdon, R. L., Weishampel, J. F., & Prince, S. (2002). Estimation of tropical forest structural characteristics using large-footprint lidar. *Remote Sensing of Environment*, *79*(2), 305–319. https://doi.org/10.1016/S0034-4257(01)00281-4

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

Guerra-Hernandez, J., & Pascual, A. (2024). *High-resolution Canopy Height Model of Hawaii Island 2018-2020*. Zenodo. https://doi.org/10.5281/ZENODO.13151991

Guerra-Hernández, J., Pascual, A., Tupinambá-Simões, F., Godinho, S., Botequim, B., Jurado-Varela, A., & Sandoval-Altelarrea, V. (2024). Using bi-temporal ALS and NFI-based time-series data to account for large-scale aboveground carbon dynamics: the showcase of mediterranean forests. *European Journal of Remote Sensing*, *57*(1), 2315413. https://doi.org/10.1080/22797254.2024.2315413

Hurlbert, S. H. (1971). The Nonconcept of Species Diversity: A Critique and Alternative Parameters. *Ecology*, *52*(4), 577–586. https://doi.org/10.2307/1934145

Kamoske, A. G., Dahlin, K. M., Stark, S. C., & Serbin, S. P. (2019). Leaf area density from airborne LiDAR: Comparing sensors and resolutions in a temperate broadleaf forest ecosystem. *Forest Ecology and Management*, *433*, 364–375. https://doi.org/10.1016/j.foreco.2018.11.017

LAStools. (2022). *Efficient LiDAR processing software*. http://rapidlasso.com/LAStools

Manning, C. (2024). *Entwine: Open source point cloud indexing*. https://entwine.io/

McElhinny, C., Gibbons, P., Brack, C., & Bauhus, J. (2005). Forest and woodland stand structural complexity: Its definition and measurement. *Forest Ecology and Management*, *218*(1), 1–24. https://doi.org/10.1016/j.foreco.2005.08.034

McGaughey, R. (2022). *FUSION/LDV: LiDAR processing and visualization software*. https://research.fs.usda.gov/pnw/products/dataandtools/tools/fusion/ldv-lidar-processing-and-visualization-software-version-4.40

Office for Coastal Management. (2024). *2018 - 2020 NOAA USGS lidar: Hawaii, HI from 2019*. NOAA National Centers for Environmental Information. https://www.fisheries.noaa.

gov/inport/item/68082

Pascual, A., Giardina, C. P., Selmants, P. C., Laramee, L. J., & Asner, G. P. (2021). A new remote sensing-based carbon sequestration potential index (CSPI): A tool to support land carbon management. *Forest Ecology and Management*, *494*, 119343. https://doi.org/10.1016/j.foreco.2021.119343

Pascual, A., & Guerra-Hernández, J. (2023). An integrated assessment of carbon emissions from forest fires beyond impacts on aboveground biomass. A showcase using airborne lidar and GEDI data over a megafire in spain. *Journal of Environmental Management*, *345*, 118709. https://doi.org/10.1016/j.jenvman.2023.118709

Pascual, A., Guerra-Hernández, J., Cosenza, D. N., & Sandoval, V. (2020). The Role of Improved Ground Positioning and Forest Structural Complexity When Performing Forest Inventory Using Airborne Laser Scanning. *Remote Sensing*, *12*(3), 413. https://doi.org/10.3390/rs12030413

Roussel, J.-R., & Auty, D. (2024). *Airborne LiDAR data manipulation and visualization for forestry applications* [Manual]. https://doi.org/10.32614/CRAN.package.lidR

Roussel, J.-R., Auty, D., Coops, N. C., Tompalski, P., Goodbody, T. R. H., Meador, A. S., Bourdon, J.-F., De Boissieu, F., & Achim, A. (2020). lidR: An R package for analysis of Airborne Laser Scanning (ALS) data. *Remote Sensing of Environment*, *251*, 112061. https://doi.org/10.1016/j.rse.2020.112061

Tang, H., & Armston, J. (2019). *Algorithm Theoretical Basis Document (ATBD) for GEDI L2B Footprint Canopy Cover and Vertical Profile Metrics*.