ARPHA Preprints

PENSOFT

Project Report

Author-formatted document posted on 05/03/2026

Published in a RIO article collection by decision of the collection editors.

DOI: https://doi.org/10.3897/arphapreprints.e190520

D4.10 Demonstrator pipeline for habitat condition metric extraction and parallel and distributed computing in a cloud environment

W. Daniel Kissling, Jinhu Wang, Yifang Shi

ARPHA Preprints *Author-formatted document posted on 05/03/2026.* DOI: https://doi.org/10.3897/arphapreprints.e190520

MAMBO
MODERN APPROACHES TO THE
MONITORING OF BIODIVERSITY

D4.10 Demonstrator pipeline for habitat condition metric extraction and parallel and distributed computing in a cloud environment

20/02/2026

Lead beneficiary: University of Amsterdam (UvA)

Authors: W. Daniel Kissling, Jinhu Wang, Yifang Shi

Reviewer: France Gerard

Funded by
the European Union

Prepared under contract from the European Commission
Grant agreement No. 101060639, EU Horizon Europe Research and Innovation Action

Project acronym:	**MAMBO**
Project full title:	**Modern Approaches to the Monitoring of Biodiversity**
Project duration:	01.09.2022 – 31.08.2026 (48 months)
Project coordinator:	Dr. Toke Thomas Høye, Aarhus University (AU)
Call:	HORIZON-CL6-2021-BIODIV-01
Deliverable title:	D4.10 Demonstrator pipeline for habitat condition metric extraction and parallel and distributed computing in a cloud environment
Deliverable n°:	D4.10
WP responsible:	WP4
Nature of the deliverable:	Demonstrator
Dissemination level:	Public
Lead beneficiary:	UvA
Due date of deliverable:	M42
Actual submission date:	M42

Deliverable status:

Version	Status	Date	Author(s)
0.1	First outline	09/02/2026	W. Daniel Kissling (UvA)
0.2	Preface, summary, appendix	10/02/2026	W. Daniel Kissling (UvA)
0.3	Input Google Colab material	10/02/2026	Jinhu Wang (UvA)
0.4	First complete draft	12/02/2026	W. Daniel Kissling (UvA)
0.5	Review	13/02/2026	France Gerard (UKCEH)
0.6	Input Demonstrator 1 &2	19/02/2026	Yifang Shi (UvA)
0.7	Figure Colab	19/02/2026	Jinhu Wang (UvA)
0.8	Final version	20/02/2026	W. Daniel Kissling (UvA)

Table of Contents

ARPHA Preprints Author-formatted document posted on 05/03/2026. DOI: https://doi.org/10.3897/arphapreprints.e190520

1 Preface

This deliverable, **D4.10 – Demonstrator pipeline for habitat condition metric extraction and parallel and distributed computing in a cloud environment**, is produced in the context of the project MAMBO (Modern Approaches to the Monitoring of BiOdiversity), funded by the European Commission through an EU Horizon Europe Research and Innovation Action (Grant Agreement No. 101060639).

The deliverable is classified as a DEM-type deliverable, meaning that its primary purpose is to **demonstrate and showcase** a functional, technical solution developed within the project, rather than to provide a comprehensive scientific or methodological report. As such, this document intentionally focuses on summarising the demonstrator pipeline, its architecture, and its application, highlighting key design choices, implementation environments, and representative use cases.

Within MAMBO, the ability to derive habitat condition indicators from **large-scale airborne LiDAR** datasets is a critical methodological challenge, due to the volume, complexity, and computational demands of national and multi-temporal point cloud data. To address this challenge, the project has developed and applied scalable workflows based on parallel, distributed, and cloud computing technologies. These workflows enable efficient extraction of vegetation structure metrics that are relevant for habitat condition assessments and biodiversity monitoring.

This deliverable provides an overview of a demonstrator pipeline built around the **Laserfarm workflow** framework and its underlying software components, as implemented and tested in different computational environments. The document highlights three complementary demonstrator implementations, ranging from high-performance computing infrastructures to cloud-based virtual research environments and lightweight notebook-based prototypes. Together, these implementations illustrate the scalability, flexibility, and transferability of the approach across different use cases, user groups, and technical settings.

In line with the demonstrator nature of this deliverable, detailed algorithmic descriptions, performance benchmarking, and scientific analyses are not repeated here, but are instead referenced through existing MAMBO deliverables, peer-reviewed publications, and publicly available code and notebooks. The emphasis of this report is on demonstrating **operational feasibility and readiness**, and on illustrating how advanced computing infrastructures can support future large-scale and harmonised habitat condition monitoring efforts in Europe.

2 Executive summary

This deliverable, **D4.10 – Demonstrator pipeline for habitat condition metric extraction and parallel and distributed computing in a cloud environment**, presents a demonstrator developed within the European Union–funded project MAMBO. It showcases how scalable, efficient, and reproducible workflows can be used to derive harmonised vegetation structure metrics from large airborne LiDAR point cloud datasets using parallel, distributed, and cloud-based computing infrastructures.

Airborne laser scanning provides detailed three-dimensional information on vegetation structure that is highly relevant for habitat condition assessment and biodiversity monitoring. However, the processing of national and multi-temporal LiDAR datasets, often consisting of many terabytes of data, poses substantial computational challenges. Within MAMBO, these challenges have been addressed through the application of the **Laserfarm workflow** framework, which builds on the Laserchicken software to compute vegetation structure metrics in a scalable and reproducible manner. Laserfarm enables the parallelisation and distributed execution of LiDAR processing tasks using modular, notebook-based workflows.

The demonstrator pipeline is presented through three complementary implementations that together illustrate different levels of scalability, accessibility, intended use, and required user expertise. The primary implementation is deployed on the Dutch national research IT infrastructure provided by SURF, using **high-throughput and cloud computing services** to process multi-terabyte airborne LiDAR datasets. This implementation has been applied both to MAMBO demonstration sites and to national-scale, multi-temporal LiDAR data for the Netherlands, demonstrating the operational feasibility of extracting vegetation structure metrics at country-wide scale using hundreds of processing cores in parallel.

As a second implementation, the Laserfarm workflow has been deployed in a **cloud-based Virtual Research Environment** of LifeWatch using NaaVRE. This demonstrator highlights how containerised workflow components and notebook-based execution can lower the barrier to entry for users, support collaborative analysis, and facilitate training and capacity building. The Virtual Research Environment approach illustrates the potential for broader uptake of LiDAR-based vegetation structure workflows across research communities and countries.

Finally, a lightweight prototype implementation is presented based on execution in **Google Colab**. While not intended for large-scale operational processing, this demonstrator illustrates how simplified LiDAR metric workflows can be executed in a fully cloud-based environment with minimal setup, supporting rapid experimentation, education, and demonstration purposes.

Together, these demonstrator pipelines show that the standardised extraction of vegetation metrics from airborne LiDAR data can be effectively scaled from local case studies to national monitoring applications, while remaining **reproducible and transferable across computing environments**. The demonstrator contributes to MAMBO by providing a practical and operational example of how advanced computing infrastructures can support harmonised, large-scale biodiversity monitoring, and it lays the foundation for future uptake and further development beyond the project.

3 List of abbreviations

ALS	Airborne Laser Scanning
ASPRS	American Society for Photogrammetry and Remote Sensing
DEM	Demonstrator, pilot, prototype, or plan design (deliverable type)
FAIR	Findability, Accessibility, Interoperability, and Reusability
EC	European Commission
ESSD	Earth System Science Data
EPSG	European Petroleum Survey Group
GDAL	Geospatial Data Abstraction Library
GeoTIFF	Geographic Tagged Image File Format
GPU	Graphics Processing Unit
HPC	High-Performance Computing
IT	Information Technology
LAS / LAZ	LiDAR data file formats (ASPRS standard)
LiDAR	Light Detection and Ranging
MAMBO	Modern Approaches to the Monitoring of BiOdiversity
NaaVRE	Notebook-as-a-Virtual Research Environment
PDAL	Point Data Abstraction Library
RAM	Random Access Memory
SLURM	Simple Linux Utility for Resource Management
SURF	Dutch national facility for information and communication technology
VRE	Virtual Research Environment

4 Scope and objectives of the demonstrator

The demonstrator presented in this deliverable addresses the need for **scalable, reproducible, and operational workflows** to derive harmonised vegetation structure metrics for habitat condition indicators from large airborne LiDAR datasets. Within the context of MAMBO, the scope of the demonstrator is to showcase how advanced computing infrastructures can be effectively combined with state-of-the-art LiDAR processing workflows to support habitat condition assessment across a wide range of spatial scales.

The demonstrator focuses on the application of the **Laserfarm** workflow framework (Kissling et al., 2022), which builds on the **Laserchicken** software (Meijer et al., 2020), to extract vegetation structure metrics from airborne laser scanning data. These metrics represent key structural characteristics of vegetation that are widely used as proxies for habitat condition in ecological and biodiversity monitoring studies (Kissling et al., 2024). The scope of this deliverable is not to introduce new indicators or algorithms, but to demonstrate how existing, well-established methods can be implemented in a way that is computationally efficient, scalable, and transferable.

Specifically, the demonstrator aims to illustrate how LiDAR processing workflows can be executed using **parallel and distributed computing**, enabling the processing of multi-terabyte, national-scale and multi-temporal point cloud datasets that would be infeasible to handle using conventional desktop-based approaches (Kissling et al., 2022; 2023; Kissling et al., 2025; Shi et al., 2025). The demonstrator therefore places strong emphasis on workflow orchestration, data partitioning, and execution across different computing environments, rather than on methodological innovation at the level of individual metrics.

The **objectives** of the demonstrator are:

- To **demonstrate scalable processing** of airborne LiDAR data, from site-level applications to national-scale analyses, using parallel and distributed computing approaches.
- To showcase the **operational use of Laserfarm workflows** for the extraction of harmonised vegetation structure metrics relevant to habitat condition assessment.
- To illustrate **portability across computing environments**, including high-performance computing (HPC) infrastructures, cloud-based services, and virtual research environments.
- To support **reproducibility and transparency**, using modular, notebook-based workflows and openly available software components.
- To **lower barriers to adoption**, by demonstrating implementations that address the needs of different user groups, ranging from infrastructure providers and national mapping agencies to researchers, practitioners, and trainees.

By focusing on these objectives, the demonstrator contributes to MAMBO by providing a concrete, technical example of how airborne LiDAR data can be transformed into vegetation structure information at scale. The demonstrator is intended to serve both as proof of concept and as a foundation for future operational uptake and further development beyond the lifetime of the project.

5 Overview of the demonstrator pipeline

5.1 Conceptual workflow

The demonstrator pipeline provides an **end-to-end workflow** for transforming large airborne LiDAR point cloud datasets into spatially explicit vegetation structure metrics (Figure 1). It is designed to be modular, scalable, and reproducible, allowing deployment across different computational infrastructures while maintaining consistent methodology (Kissling et al., 2022).

The pipeline builds on the **Laserfarm framework** (Kissling et al., 2022) and the underlying **Laserchicken software** (Meijer et al., 2020) to convert raw three-dimensional airborne laser scanning (ALS) data into harmonised raster layers of vegetation structure metrics suitable for habitat condition assessment.

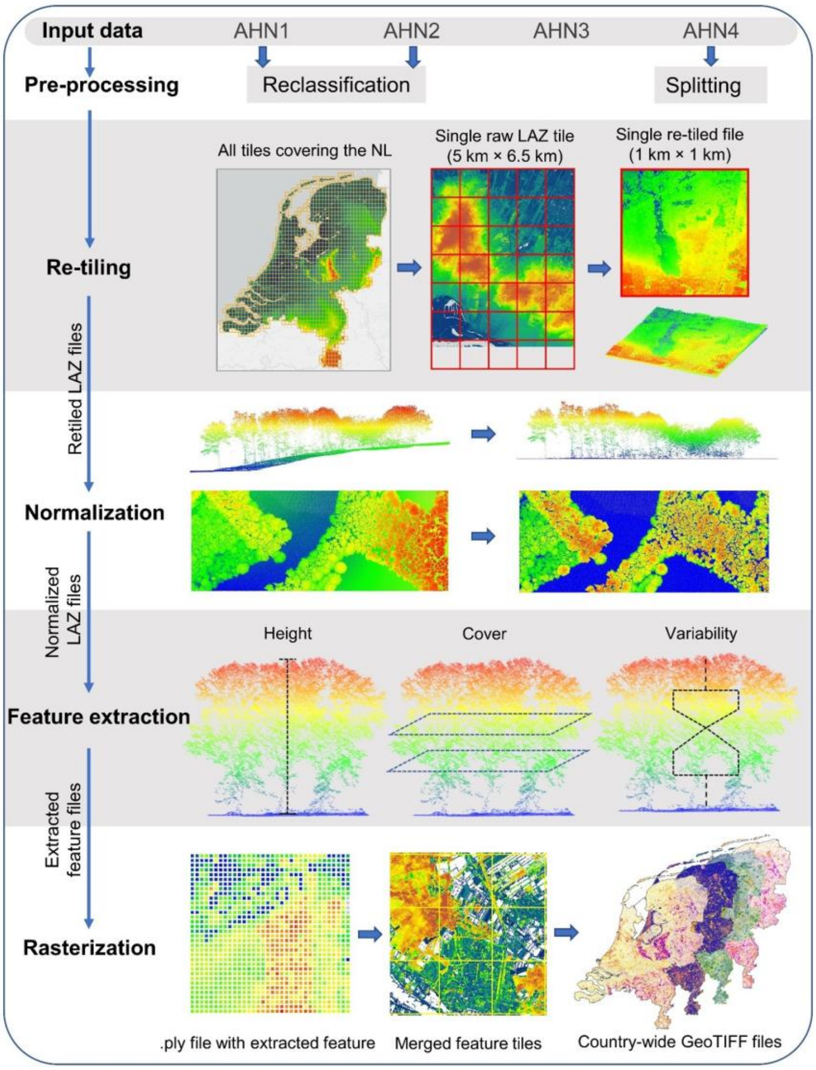


Figure 1: Overview of the demonstrator pipeline applied to multi-terabyte, national-scale, and multi-temporal airborne LiDAR point cloud datasets in the Netherlands. Following data ingestion and pre-processing, the four core modules of the Laserfarm workflow—re-tiling, normalization, feature extraction, and rasterization—are executed in a scalable, distributed computing environment to generate harmonised raster layers of vegetation structure metrics. Adapted from (Shi et al., 2025).

Conceptually, the workflow comprises four main stages:

1. **Data ingestion and pre-processing**

   Raw ALS point cloud files (LAS/LAZ format) are accessed from national or regional repositories. For site-based applications, datasets are clipped to defined boundaries (e.g. protected areas) (Kissling et al., 2025), while for national-scale the processing is wall-to-wall (Kissling et al., 2023; Shi et al., 2025). Pre-processing steps optimise data organisation and prepare datasets for efficient large-scale computation (Shi et al., 2025).

2. **Data partitioning and parallelisation**

   To enable scalable handling of multi-terabyte datasets, point clouds are subdivided into spatial tiles ('Re-tiling' in Figure 1). These independent units allow distributed execution across multiple compute nodes, where identical processing routines are performed in parallel. This spatial partitioning forms the basis for efficient high-throughput computation (Kissling et al., 2022).

3. **Normalization and feature extraction**

   LiDAR return heights are normalised relative to the terrain surface to derive vegetation height above ground ('Normalization' in Figure 1). Subsequently, a harmonised set of vegetation structure metrics is calculated for predefined spatial units ('Feature extraction' in Figure 1), typically 10 m × 10 m grid cells. These metrics describe vegetation height distribution, canopy cover and openness (e.g. pulse penetration ratio), and vertical structural variability (Kissling et al., 2022; 2023; Kissling et al., 2025; Shi et al., 2025). Consistent metric definitions and calculations are applied across datasets with differing point densities and acquisition parameters, supporting comparability across regions and time periods (see robustness and sensitivity analyses in Kissling et al., 2024; Shi et al., 2025).

4. **Rasterization and export of geospatial products**

   Computed metrics are aggregated and exported as spatially continuous raster layers (GeoTIFF format) using standard coordinate reference systems ('Rasterization' in Figure 1). The resulting wall-to-wall products are suitable for habitat condition assessment, ecological analyses, and publication in open repositories (Kissling et al., 2023; Kissling et al., 2025; Shi et al., 2025).

A central feature of the conceptual workflow is the separation between methodological logic and computational infrastructure. The same processing sequence can be executed on HPC clusters, cloud platforms, or virtual research environments without altering the underlying analytical approach (Kissling et al., 2022; Wang et al., 2022; Zhao et al., 2022; Shi et al., 2025). This portability ensures methodological stability while allowing computational resources to be adapted to the scale and complexity of the input data.

The workflow is implemented through Jupyter Notebooks and open-source software components, supporting transparency, traceability, and reproducibility. By combining modular design with distributed execution, the demonstrator provides a scalable and transferable solution for deriving harmonised vegetation structure metrics from airborne LiDAR data across local, national, and transnational contexts.

5.2 Core software components

The demonstrator pipeline is built on modular, open-source software components that enable distributed and reproducible processing of large airborne LiDAR datasets. The main components are the Laserfarm workflow framework, the Laserchicken software, and Jupyter Notebook–based workflow implementation.

Laserfarm workflow framework

Laserfarm provides the orchestration layer of the pipeline (Kissling et al., 2022). It structures the processing chain into four modules—re-tiling, normalization, feature extraction, and rasterization—which can be executed across spatial partitions of the input data.

A core design principle of Laserfarm is scalable parallel processing (Kissling et al., 2022). Large point cloud datasets are partitioned into spatial tiles that can be processed concurrently across multiple workers in a compute cluster or cloud environment. This distributed architecture enables efficient handling of multi-terabyte datasets and billions of LiDAR returns while ensuring consistent calculation of vegetation structure metrics. The modular workflow design further allows adaptation to different study areas, heterogeneous ALS datasets, and diverse IT infrastructures without altering the underlying analytical logic.

Laserfarm is implemented in Python (available via PyPI: https://pypi.org/project/laserfarm/) and integrates with distributed computing libraries, such as Dask-based environments, to manage task scheduling, worker allocation, and memory usage (Kissling et al., 2022). By separating workflow logic from infrastructure configuration, it enables deployment across high-performance computing systems, cloud platforms, and virtual research environments. This infrastructure-agnostic design allows the workflow to leverage available distributed computing resources efficiently, thereby maximising scalability and operational flexibility.

Laserchicken software

Laserchicken (Meijer et al., 2020) performs the calculation of LiDAR-derived metrics. It provides efficient algorithms for extracting statistical and structural properties from three-dimensional point clouds within user-defined spatial units. Source code is in Python and available on GitHub (https://github.com/eEcoLiDAR/laserchicken).

Supported metrics include vegetation height percentiles, measures of canopy density and openness, indices of vertical variability, and geometric descriptors derived from neighbourhood analysis (Meijer et al., 2020). The software can process specific subsets of points (e.g. vegetation-only classes or all returns), allowing flexible metric computation while maintaining methodological consistency across applications.

Jupyter Notebooks and workflow reproducibility

Execution of the workflow is implemented through Jupyter Notebooks, which combine code, parameter settings, and documentation in a transparent and shareable format. MAMBO implementations are publicly available via GitHub (e.g. https://github.com/ShiYifang/AHN) and WorkflowHub (e.g. https://workflowhub.eu/projects/302#workflows).

The notebook-based design enhances reproducibility by explicitly documenting input/output paths, coordinate reference systems, tiling grids, and metric definitions. While infrastructure-specific notebooks require adaptation of configuration settings (e.g. resource allocation, task scheduling, and storage paths), the underlying analytical processes and conceptual workflow steps remain consistent across implementations (Kissling et al., 2022; Wang et al., 2022; Zhao et al., 2022; Shi et al., 2025). In all cases, the same modular logic—based on Laserfarm and Laserchicken—is preserved, enabling deployment on HPC clusters, cloud-based virtual research environments, and lightweight platforms with appropriate infrastructure-specific adjustments.

Together, the workflow logic and implementation design underlying Laserfarm and Laserchicken—along with the notebook-based implementations—form the technical backbone of the demonstrator pipeline. While not all demonstrators directly use the Laserfarm and Laserchicken software (e.g. the Google Colab prototype), they follow the same modular processing logic and conceptual workflow steps. This integrated design enables the scalable and reproducible transformation of raw ALS point clouds into harmonised vegetation structure metrics for habitat condition assessment, from site-level applications to national monitoring programmes.

6 Demonstrator implementations

6.1 Demonstrator 1: High-performance and cloud computing pipeline on SURF

The primary demonstrator implementation of the vegetation metric extraction pipeline was deployed on the Dutch national research IT infrastructure provided by SURF (https://www.surf.nl/en). This represents the flagship, production-level deployment of the workflow and demonstrates its capacity to process multi-terabyte, national-scale, and multi-temporal airborne LiDAR datasets within an operational HPC environment.

Infrastructure and computational environment

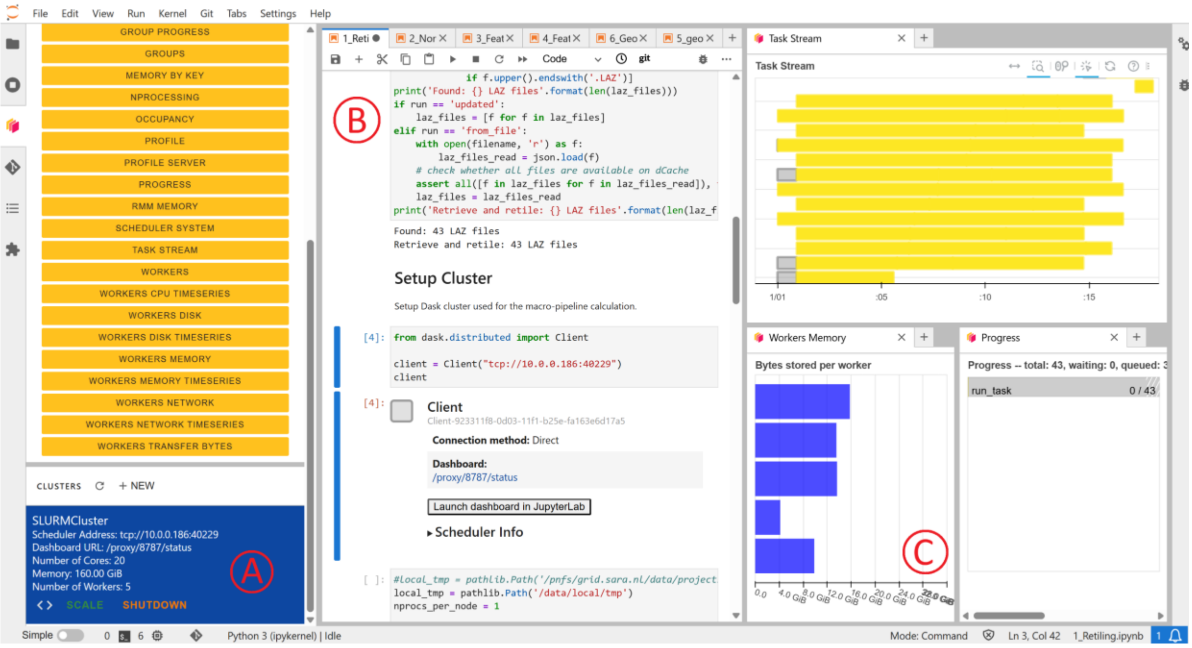
Processing was conducted on SURF's high-throughput data processing platform Spider (https://doc.spider.surfsara.nl/en/latest/), an in-house compute cluster designed for large-scale, data-intensive workloads. Spider operates as a SLURM-managed HPC cluster, supporting distributed job scheduling and large-scale parallel execution across customised compute nodes and clusters. Distributed task scheduling was orchestrated using Dask (https://www.dask.org/), an open-source Python library for parallel and distributed computing, enabling the simultaneous utilisation of many hundreds of CPU cores (Figure 2).

The computational setup combined:

- Distributed storage for large LiDAR point cloud datasets
- Parallel execution across multiple worker nodes
- Flexible resource allocation tailored to dataset size and processing demands
- Integration with Jupyter-based workflow environments and Dask-driven distributed task scheduling

Together, this infrastructure provided the capacity to process billions of LiDAR points efficiently while maintaining transparent and reproducible workflow execution.

Prior to large-scale processing, raw point clouds were systematically inspected to verify classification schemes, bounding boxes, and spatial extents. This preparatory step ensured correct configuration of re-tiling grids, spatial partitioning parameters, and class filtering before distributed execution. Processing was conducted within controlled virtual environments with fixed software dependencies, ensuring consistent behaviour across repeated runs and across different computational infrastructures.



SLURMCluster
Scheduler Address: tcp://10.0.0.186:40229
Dashboard URL: /proxy/8787/status
Number of Cores: 20
Memory: 160.00 GiB
Number of Workers: 5

```
                if f.upper().endswith('.LAZ')]
print('Found: {} LAZ files'.format(len(laz_files)))
if run == 'updated':
    laz_files = [f for f in laz_files]
elif run == 'from_file':
    with open(filename, 'r') as f:
        laz_files_read = json.load(f)
    # check whether all files are available on dCache
    assert all([f in laz_files for f in laz_files_read]),
    laz_files = laz_files_read
print('Retrieve and retile: {} LAZ files'.format(len(laz_f
```

```
Found: 43 LAZ files
Retrieve and retile: 43 LAZ files
```

Setup Cluster

Setup Dask cluster used for the macro-pipeline calculation.

```
[4]: from dask.distributed import Client

     client = Client("tcp://10.0.0.186:40229")
     client
```

[4]: Client
Client-923311f8-0d03-11f1-b25e-fa163e6d17a5

Connection method: Direct

Dashboard: /proxy/8787/status

Launch dashboard in JupyterLab

▸ Scheduler Info

```
[ ]: #local_tmp = pathlib.Path('/pnfs/grid.sara.nl/data/project
     local_tmp = pathlib.Path('/data/local/tmp')
     nprocs_per_node = 1
```

Progress -- total: 43, waiting: 0, queued: 3

Python 3 (ipykernel) | Idle

Figure 2: Distributed processing of national-scale ALS point clouds in the Netherlands using services of the Dutch national research IT infrastructure SURF, based on Jupyter, Dask, and a SLURM-managed HPC cluster. Panel A shows the scalable SLURM cluster configuration, enabling dynamic allocation of worker nodes according to workflow demands. Panel B presents a Laserfarm workflow module implemented as a Jupyter Notebook for tile-based processing. Panel C displays the Dask dashboard for real-time monitoring of parallel task execution, where each yellow bar represents a successfully processed input tile.

Applications within MAMBO

The SURF-based pipeline was applied to two major use cases within the MAMBO project:

1. **MAMBO demonstration sites across Europe**

   National or regional ALS datasets were clipped to site boundaries and processed using the Laserfarm workflow (Kissling et al., 2025). The HPC environment enabled rapid and standardised computation of vegetation structure metrics across heterogeneous LiDAR datasets originating from multiple countries and national repositories.

2. **National-scale, multi-temporal LiDAR metrics for the Netherlands**

   The workflow was applied to successive country-wide ALS campaigns (AHN1–AHN4), covering the entire Netherlands over multiple decades (Shi et al., 2025). These datasets comprise multi-terabyte point clouds containing hundreds of billions of returns. The SURF infrastructure enabled distributed re-tiling, normalization, feature extraction, and rasterization at 10 m spatial resolution across the full national extent.

This demonstrator illustrates the feasibility of consistent, large-scale vegetation structure monitoring using repeated national LiDAR surveys. The full processing configuration for processing the AHN datasets on the SURF infrastructure is publicly documented in a dedicated repository (https://doi.org/10.5281/zenodo.15579063), further supporting transparency and reproducibility of the national-scale implementation.

Demonstrated scalability and performance

The SURF implementation highlights several key aspects of scalability:

- **Parallel processing of spatial tiles**, enabling simultaneous computation across independent grid units
- **Efficient memory management**, ensuring that tile sizes remained within worker memory limits
- **Flexible resource allocation**, allowing the number of workers and cores to be adapted to dataset size and processing steps
- **Reproducible execution**, with identical workflow logic applied across demonstration sites and national datasets

By decoupling workflow logic from infrastructure configuration, the same Laserfarm notebooks were executed on SURF without modification of methodological steps. Only environment-specific parameters (e.g. cluster configuration, storage paths, and resource allocation settings) and workflow parameters related to the retiling grid and ASPRS classification value required adaptation.

The workflow further supports fault-tolerant execution by logging tile-level processing status and enabling selective re-execution of failed tiles. This minimises unnecessary recomputation and enhances resource efficiency in large-scale distributed environments. In addition, detailed logs for each processed file and workflow step provide traceability and serve as a metadata record of the execution process.

Operational relevance

The SURF-based demonstrator represents a near-operational implementation of large-scale LiDAR-based vegetation structure monitoring. It demonstrates that:

- Multi-temporal national ALS datasets with a multi-terabyte data volume (~70 TB of uncompressed raw point clouds) can be processed consistently using distributed computing
- Harmonised vegetation structure metrics can be derived at country-wide scale
- High-performance and cloud infrastructures can support routine, repeatable metric generation

This implementation provides a blueprint for research infrastructures, national mapping agencies, and environmental monitoring organisations seeking to operationalise habitat condition assessment based on vegetation structure metrics. It demonstrates that advanced computational environments are not merely experimental tools, but can support scalable, reproducible, and policy-relevant biodiversity monitoring workflows.

6.2 Demonstrator 2: Virtual Research Environment pipeline on NaaVRE

In addition to the HPC deployment on SURF, the Laserfarm workflow was implemented in a cloud-based Virtual Research Environment (VRE) using NaaVRE (Notebook-as-a-VRE) (Zhao et al., 2022). This demonstrator illustrates how scalable LiDAR processing workflows can be made accessible through browser-based, containerised research environments, thereby lowering technical barriers while preserving reproducibility and computational robustness.

Virtual Research Environment concept

NaaVRE (https://naavre.net/) provides a collaborative cloud infrastructure developed within LifeWatch (https://www.lifewatch.eu/) in which Jupyter-based workflows can be executed within pre-configured, containerised environments. Users access the environment through a web interface (https://beta.naavre.net/vreapp/vl/laserfarm), without requiring local software installation or direct interaction with HPC command-line systems. Computational tasks can be executed on underlying cluster resources or connected cloud infrastructure, depending on configuration and resource availability.

Within this framework, the Laserfarm workflow was deployed within a dedicated “LiDAR Virtual Lab”, building on the open-source configuration available in the NaaVRE LiDAR Virtual Lab repository (https://github.com/NaaVRE/vl-laserfarm). The environment encapsulates all required dependencies (e.g. PDAL, GDAL, Laserfarm, Laserchicken) within a controlled container setup. This ensures consistent execution behaviour across users and sessions while avoiding dependency conflicts that may arise in locally configured environments.

Implementation of Laserfarm within NaaVRE

The Laserfarm modules—re-tiling, normalization, feature extraction, and rasterization—were integrated into NaaVRE as executable notebook-based components. Within the Virtual Lab, users can configure and compose workflows using the experiment manager, where containerised processing cells are assembled into a structured execution pipeline (Figure 3).

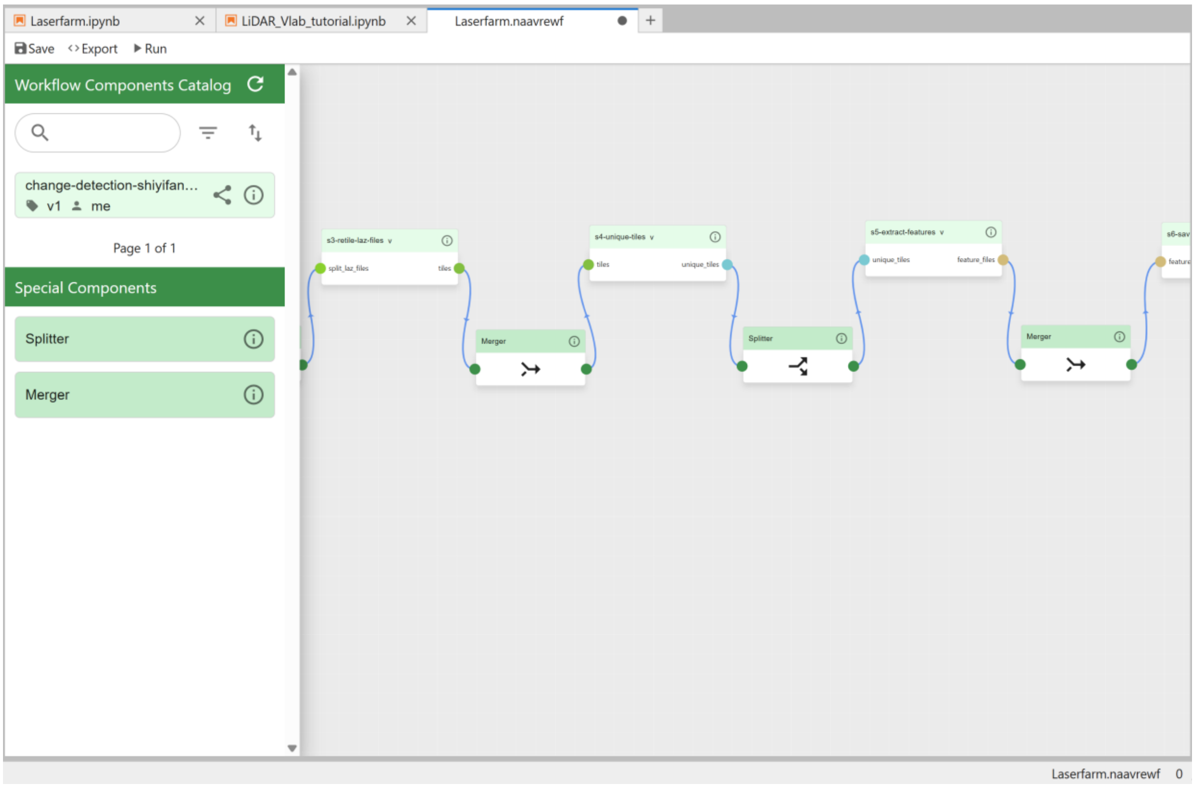
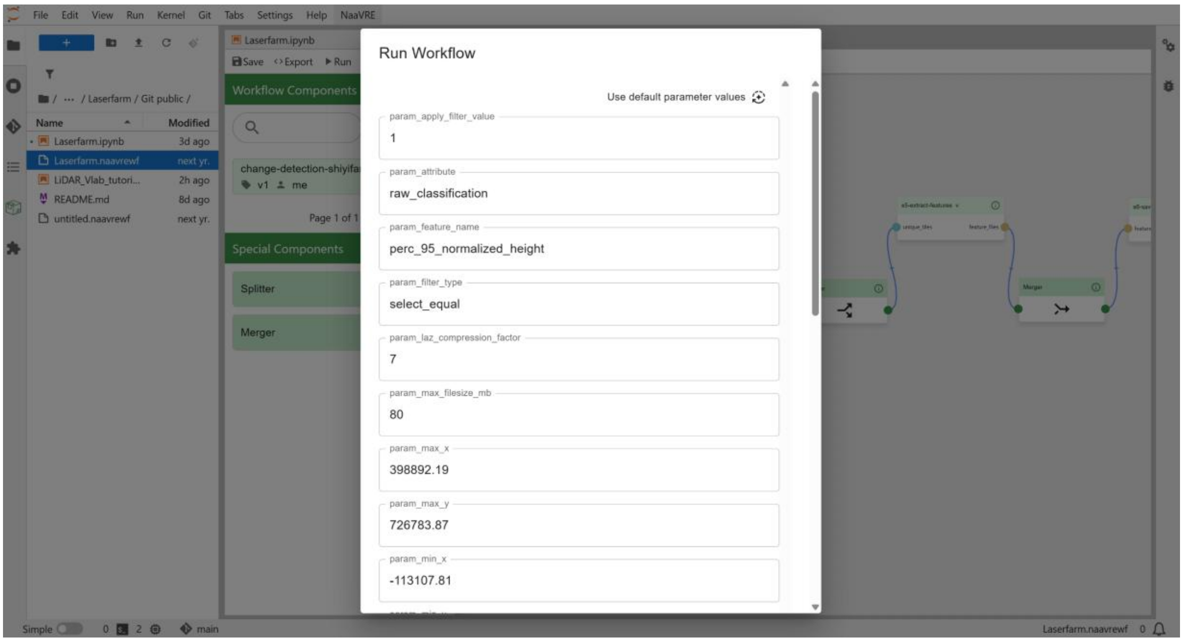


Figure 3: NaaVRE workflow manager interface for the Laserfarm workflow. Each Jupyter Notebook module is encapsulated as a containerised component within the workflow. Dedicated components such as ‘Splitter’ and ‘Merger’ enable parallel task distribution and subsequent consolidation of outputs.

During workflow composition, dedicated components such as ‘Splitter’ and ‘Merger’ enable parallel execution by distributing tasks across spatial partitions and subsequently consolidating results (Figure 3). This design allows distributed processing while abstracting the underlying infrastructure complexity from the user.

Configuration files and parameter settings—including tiling grids, feature definitions, spatial resolution, class filtering options, and coordinate reference systems—are defined within the notebook environment prior to execution (Figure 4). This approach ensures transparency and facilitates adaptation of the workflow to different datasets and study areas while preserving the underlying analytical logic.



Run Workflow

Use default parameter values

param_apply_filter_value: 1
param_attribute: raw_classification
param_feature_name: perc_95_normalized_height
param_filter_type: select_equal
param_laz_compression_factor: 7
param_max_filesize_mb: 80
param_max_x: 398892.19
param_max_y: 726783.87
param_min_x: -113107.81

Figure 4: User-defined parameters and configuration settings within the NaaVRE Virtual Lab, including feature selection, area of interest, and input data characteristics. These settings are specified prior to execution, enabling users to customise and run the workflow without directly modifying the underlying Python source code.

The NaaVRE implementation preserves the same methodological logic as the SURF deployment while abstracting infrastructure-specific complexity. Users interact with:

- Pre-configured notebook templates
- Containerised execution environments
- Integrated data access, storage, and compute resources
- Virtual Lab–specific parallelisation components
- An Argo workflow interface for real-time monitoring of workflow execution (Figure 5)

This architecture demonstrates that the Laserfarm workflow can be executed in a cloud-based environment without requiring users to manually configure SLURM clusters, virtual environments, or distributed scheduling frameworks. At the same time, backend orchestration ensures that moderate-scale datasets can still be processed efficiently using distributed computation.

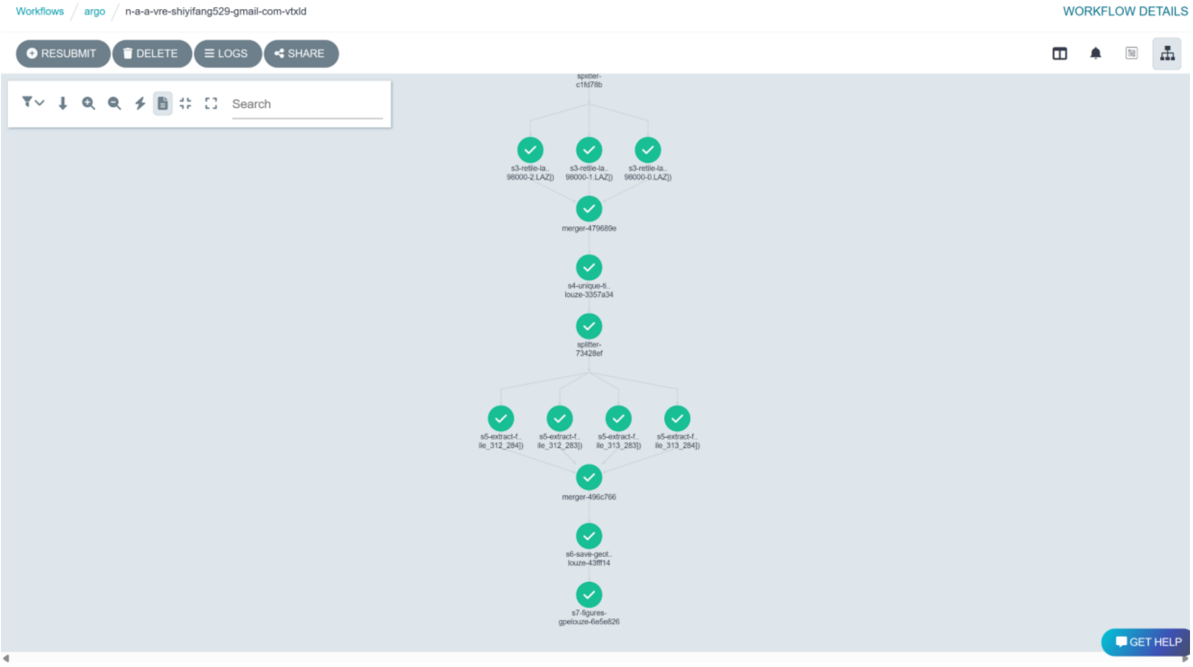


Figure 5: Argo workflow interface displaying the real-time execution status of the submitted workflow. Successfully completed steps are indicated by green checkmarks. The inclusion of a 'Splitter' component enables subsequent processing steps to run in parallel.

Use within MAMBO and training activities

Within MAMBO, the NaaVRE deployment currently serves as a prototype environment primarily used by researchers from Work Package 4 (WP4). Its use to date has focused on testing and validating the Laserfarm workflow in a cloud-based setting:

1. **Prototype research environment for vegetation structure metric extraction**

   Researchers from WP4 have implemented and executed the Laserfarm workflow within NaaVRE to evaluate its functionality, reproducibility, and portability beyond a dedicated HPC infrastructure. The deployment has so far been restricted to internal project use. Broader adoption would facilitate experimentation, cross-institutional collaboration, and reproducible analysis without requiring direct access to national high-performance computing facilities.

2. **Training and capacity building**

   The Virtual Lab is intended for use in training and knowledge transfer activities, including the MAMBO workshop on large-scale and scalable processing of airborne LiDAR for vegetation structure analysis (Work Package 5). Dedicated training materials and tutorial notebooks are available through the NaaVRE LiDAR Virtual Lab repository on GitHub (https://github.com/NaaVRE/vl-laserfarm/blob/main/LiDAR_Vlab_tutorial.ipynb). The containerised environment ensures that participants work within identical, pre-configured software setups, eliminating installation barriers and minimising technical troubleshooting during hands-on sessions. This facilitates efficient onboarding and supports capacity building in scalable LiDAR processing methods.

Added value compared to HPC-only deployment

The NaaVRE demonstrator complements the SURF implementation by emphasising accessibility, portability, and collaborative use. Its key advantages include:

- Browser-based access without local software installation
- Containerised environments that ensure reproducibility
- Reduced technical barriers compared to direct HPC cluster usage
- Simplified workflow configuration and execution
- Support for collaborative development and sharing of workflows

While processing of large national ALS datasets benefit from dedicated HPC infrastructures, the NaaVRE implementation demonstrates how scalable workflows for LiDAR-based vegetation structure extraction can be made accessible to a broader European research community, for example through research infrastructure services provided by LifeWatch.

Operational and strategic relevance

The VRE demonstrator highlights the importance of cloud-based research infrastructures for biodiversity monitoring. By encapsulating advanced LiDAR processing workflows within managed, reproducible environments, the approach supports:

- FAIR and open science practices
- Cross-border collaboration
- Capacity building across institutions
- Reduced technical entry barriers for ecological remote sensing applications

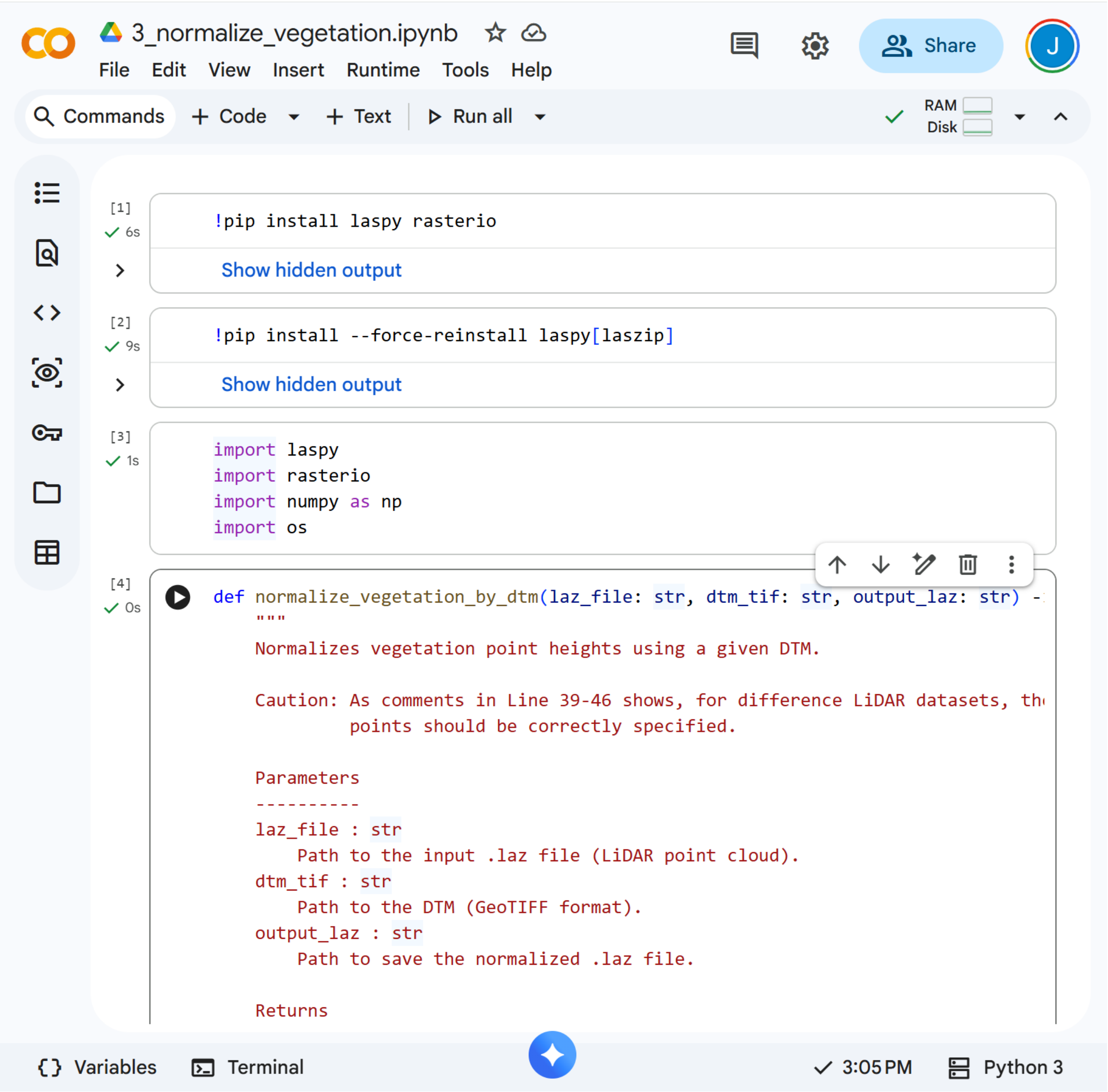
Together with the SURF-based HPC implementation, the NaaVRE deployment demonstrates that the Laserfarm workflow is not tied to a single infrastructure but can be adapted to different computational ecosystems. This flexibility strengthens its potential for long-term uptake within European Research Infrastructures such as LifeWatch ERIC (https://www.lifewatch.eu/) or eLTER RI (https://elter-ri.eu/), thereby facilitating European biodiversity monitoring initiatives beyond the lifetime of MAMBO.

6.3 Demonstrator 3: Lightweight prototype pipeline on Google Colab

In addition to the HPC and VRE implementations, a lightweight prototype of the vegetation structure metric extraction workflow was developed for execution on Google Colab (https://colab.research.google.com/). While both NaaVRE and Google Colab provide browser-based cloud environments, this demonstrator represents a simplified, standalone notebook implementation that does not rely on containerised Virtual Lab infrastructure. It illustrates how LiDAR-based vegetation structure metrics can be computed in a fully cloud-based setting without requiring access to dedicated HPC systems or managed research infrastructures.

Concept and scope

The Colab-based prototype was designed to provide an accessible, self-contained implementation of LiDAR vegetation structure metric extraction for small to moderate-sized ALS datasets. While it does not directly use the Laserfarm workflow and Laserchicken software libraries, it follows a comparable modular workflow structure and implements equivalent vegetation metric calculations to ensure standardised and conceptually consistent processing. The workflow is implemented in Python and structured as a Jupyter Notebook to support "literate programming", where executable code is combined with narrative documentation and interactive visualisation (Figure 6).



```
!pip install laspy rasterio
```

```
!pip install --force-reinstall laspy[laszip]
```

```
import laspy
import rasterio
import numpy as np
import os
```

```
def normalize_vegetation_by_dtm(laz_file: str, dtm_tif: str, output_laz: str) -
    """
    Normalizes vegetation point heights using a given DTM.

    Caution: As comments in Line 39-46 shows, for difference LiDAR datasets, th
             points should be correctly specified.

    Parameters
    ----------
    laz_file : str
        Path to the input .laz file (LiDAR point cloud).
    dtm_tif : str
        Path to the DTM (GeoTIFF format).
    output_laz : str
        Path to save the normalized .laz file.

    Returns
```

Figure 6: Example of a Jupyter Notebook running on Google Colab, illustrating the setup of a vegetation point normalization step. Executing the initial cells installs and imports the required dependencies into the runtime environment. Subsequent cells contain the processing functions that perform the defined workflow operations.

The processing logic follows the same four-step structure as the Laserfarm-based pipeline:

1. **Pre-processing and spatial tiling**

   Raw LAS/LAZ point cloud data are ingested and partitioned into manageable spatial tiles to ensure efficient memory usage during processing. This tile-based approach prevents memory overflow and allows the workflow to operate within the limited RAM resources typically available in cloud notebook environments. During this step, users can define the tile dimensions through configurable parameters, enabling adaptation to dataset size and computational constraints.

2. **Height normalization**

   Vegetation point elevations are normalised relative to terrain height derived from ground-classified points, converting absolute elevation into vegetation height above ground level.

3. **Metric calculation**

   A set of 25 vegetation structure metrics is computed, covering:

   - Vegetation height statistics (e.g. maximum and percentile-based descriptors)
   - Vegetation cover and density metrics
   - Vertical structural variability indices (e.g. entropy and kurtosis of height distributions)

4. **Rasterization and export**

   The calculated metrics are aggregated into spatial grids and exported as GeoTIFF raster products at a predefined spatial resolution, enabling integration with other geospatial datasets.

Implementation characteristics

The Colab implementation is publicly available via GitHub (https://github.com/Jinhu-Wang/LiDAR_Vegetation_Metrics) and supports dual deployment: execution within Google Colab notebooks or as standalone Python scripts on local machines. In the Colab configuration, users need to mount a Google Drive for data access and storage, and required dependencies are installed dynamically within the temporary runtime environment.

Several characteristics of this demonstrator overlap with the NaaVRE Virtual Lab approach, particularly in terms of accessibility and cloud-based execution. These include:

- Fully browser-based execution without local installation
- Automated environment setup within the runtime environment
- Tile-based memory management adapted to constrained computational resources

In contrast to the containerised and orchestrated execution in NaaVRE, the Colab implementation operates as a lightweight, standalone notebook environment. It additionally provides interactive visualisation of 3D point clouds and derived raster metrics to support quality control and exploratory analysis.

While Colab offers free access to cloud compute resources, it is subject to session timeouts, limited memory capacity, and restricted runtime duration. Consequently, this demonstrator is not intended for national-scale or multi-terabyte datasets, but rather for prototyping, teaching, and experimentation with smaller LiDAR subsets.

Role within the demonstrator portfolio

The Google Colab prototype complements the SURF-based HPC implementation and the NaaVRE Virtual Lab by addressing a distinct segment of the user and infrastructure spectrum:

- SURF – large-scale, operational deployment on high-performance computing infrastructure for expert users
- NaaVRE – containerised, collaborative cloud execution for research and training environments
- Google Colab – low-barrier, lightweight cloud prototyping for experimentation and teaching

This layered demonstrator strategy illustrates that LiDAR-based vegetation structure metric workflows can be adapted to different computational capacities and levels of user expertise, ranging from exploratory analysis and capacity building to national-scale operational monitoring.

Operational and educational relevance

Although limited in scalability, the Colab-based demonstrator plays an important role in lowering entry barriers to LiDAR-based vegetation structure analysis. It enables:

- Rapid testing of workflows without dedicated infrastructure setup
- Teaching and demonstration of vegetation metric computation principles
- Reproducible sharing of executable notebooks through public repositories
- Exploratory data analysis prior to migration to HPC environments
- Derivation of site-specific vegetation structure metrics (e.g. for nature reserves or local study areas)

By providing an accessible entry point to LiDAR vegetation structure metric extraction, the lightweight Colab prototype supports capacity building and expands the potential user community for scalable vegetation structure monitoring workflows.

7 Comparison of demonstrator pipelines

The three demonstrator implementations developed within MAMBO show how the vegetation structure metric extraction workflow can be deployed across diverse computational environments, each addressing different user skill levels, data volumes, and operational requirements. Together, they form a complementary portfolio that demonstrates both portability and scalability, ranging from lightweight, site-level prototyping to national-scale, production-level processing.

7.1 Overview of key characteristics

The main characteristics of the three demonstrator pipelines are summarised below, covering infrastructure type, primary objective, data scale, parallelisation, software environment, target users, and operational maturity (Table 1).

7.2 Scalability gradient

A central outcome of the demonstrator portfolio is the clear scalability gradient across implementations:

- **SURF-based pipeline**: Demonstrates full scalability for national and multi-temporal airborne LiDAR archives comprising tens of terabytes of data. It supports distributed execution across hundreds of CPU cores and is suitable for operational biodiversity monitoring at national scale.
- **NaaVRE implementation**: Maintains the same analytical workflow while reducing infrastructure complexity. It enables containerised, browser-based execution and collaborative use, making it well suited for cross-institutional research and training. While the Virtual Lab supports distributed processing in principle, large-scale operational deployment is currently constrained by the availability of sufficient computational resources, large-scale storage capacity, a harmonised metadata

catalogue covering national and regional ALS datasets across Europe, and a centralised data repository for processing heterogeneous national LiDAR archives.

- **Google Colab prototype**: Prioritises accessibility and rapid deployment over computational scale. It allows execution in a lightweight browser-based environment but is constrained by session limits and memory availability. It is therefore best suited for experimentation, demonstration, and teaching.

This gradient demonstrates that the underlying workflow logic is portable across infrastructures, while available computational resources determine the achievable scale and operational scope of application.

Table 1: Comparison of the three demonstrator pipelines.

Aspect	Demonstrator 1: SURF (HPC/Cloud)	Demonstrator 2: NaaVRE (VRE)	Demonstrator 3: Google Colab
Infrastructure type	National high-performance computing and cloud infrastructure	Cloud-based Virtual Research Environment	Public cloud notebook platform
Primary objective	Operational large-scale processing	Accessible, containerised collaborative execution	Lightweight prototyping and teaching
Data scale	Multi-terabyte, national and multi-temporal datasets	Moderate to large datasets	Datasets within Google Drive storage, URLs, cloud storages, Dropbox and OneDrive storages
Parallelisation	Distributed execution via SLURM and Dask across compute nodes	Backend-managed distributed scheduling	Limited parallelisation within Colab runtime
Software environment	Controlled virtual environments on HPC	Containerised execution environments	Ephemeral cloud runtime with notebook-based setup
Target users	Expert users with advanced HPC and scripting skills	Researchers, practitioners, and students without required programming expertise	Users with basic Python knowledge seeking flexible notebook-based experimentation
Operational maturity	Production-level implementation	Prototype (internal WP4 use)	Proof-of-concept prototype

7.3 Complementarity of the implementations

The three demonstrators represent complementary solutions rather than competing alternatives, each addressing a different stage of workflow adoption:

1. **Exploration and learning (Colab)** – Enables experimentation with vegetation structure metric extraction on small datasets in a lightweight, low-barrier environment.
2. **Collaborative research and training (NaaVRE)** – Supports shared, containerised execution with simplified configuration, facilitating cross-institutional collaboration and capacity building without requiring programming expertise.

3. **Operational monitoring (SURF)** – Enables efficient and reproducible processing of large-scale, multi-temporal LiDAR datasets at national scale using high-performance computing resources.

Together, this layered structure establishes a progressive adoption pathway: users can begin with exploratory analysis, advance to collaborative and structured research environments, and ultimately transition to production-level processing as data volumes and analytical demands increase.

7.4 Reproducibility and portability

Across all three implementations, the core methodological components—re-tiling, normalization, feature extraction, and rasterization—remain conceptually identical. While infrastructure-specific configurations differ, the underlying analytical logic and metric definitions are preserved.

Reproducibility is supported through:

- Open-source software components
- Jupyter notebooks and repositories
- Explicit parameterisation of spatial grids, classification filters, and coordinate reference systems
- Containerisation (NaaVRE) and controlled virtual environments (SURF) to ensure consistent software dependencies

Together, these elements demonstrate that vegetation structure metric extraction is not tied to a specific computational platform but can be deployed across diverse infrastructures while maintaining methodological consistency.

7.5 Strategic implications

From a strategic perspective, the comparison of demonstrator pipelines shows that scalable LiDAR-based vegetation structure monitoring can be:

- Operationally implemented at national scale through high-performance computing (HPC deployment)
- Made broadly accessible through containerised virtual research environments (VRE deployment)
- Applied in education and exploratory settings through lightweight cloud prototypes (Colab prototype)

By validating the workflow across these complementary contexts, MAMBO demonstrates both technical maturity and infrastructural flexibility. This enhances the potential for long-term integration within European research infrastructures and environmental monitoring frameworks, while promoting capacity building, open science, and methodological transparency.

8 Reusability, scalability, and transferability

The demonstrator pipeline was designed to ensure reusability across projects, scalability across data volumes, and transferability across infrastructures, geographic contexts, and spatial extents. These dimensions support long-term uptake of LiDAR-based vegetation structure metric extraction beyond MAMBO.

8.1 Reusability

Reusability is ensured through open-source software, modular workflow design, and publicly accessible repositories. The workflow is implemented in Python and executed via Jupyter Notebooks, where parameters, spatial configurations, and processing steps are explicitly documented.

Tutorials and example implementations are available for:

- **HPC processing** on national research IT infrastructures such as SURF in the Netherlands (https://github.com/ShiYifang/AHN4_on_Spider)
- **Cloud-based processing** using NaaVRE within the LiDAR Virtual Lab of LifeWatch (https://github.com/NaaVRE/vl-laserfarm/blob/main/LiDAR_Vlab_tutorial.ipynb)
- **Lightweight cloud execution** via Google Colab (https://github.com/Jinhu-Wang/LiDAR_Vegetation_Metrics)

Key elements supporting reusability include:

- Modular workflow structure (re-tiling, normalization, feature extraction, rasterization)
- Explicit configuration of spatial grids, coordinate reference systems, and classification filters
- Public repositories providing code, documentation, and executable notebooks
- Containerised (NaaVRE) and controlled virtual environments (SURF) for consistent dependency management

The Google Colab prototype further lowers entry barriers by enabling experimentation without specialised infrastructure. Together, these elements ensure that the workflow can be inspected, reproduced, and adapted by other institutions and research communities.

8.2 Scalability

Scalability has been demonstrated across three infrastructure levels:

1. **HPC (SURF)**: Processing of multi-terabyte national datasets containing hundreds of billions of LiDAR returns.
2. **Virtual Research Environment (NaaVRE)**: Distributed execution of moderate-scale datasets within containerised, browser-based environments.
3. **Google Colab**: Tile-based processing of smaller datasets within constrained cloud notebook runtimes.

The core workflow logic is independent of infrastructure scale. Spatial tiling and distributed task scheduling provide the foundation for horizontal scalability, allowing computational resources to be aligned with dataset size and processing requirements.

8.3 Transferability

Transferability has been demonstrated across heterogeneous airborne LiDAR datasets from multiple European countries. Variations in point density, acquisition period, classification schemes, and coordinate systems are accommodated through configurable workflow parameters.

Transferability is supported by:

- Standard LAS/LAZ file formats
- Widely used coordinate reference systems
- Flexible ASPRS class filtering
- Harmonised metric definitions independent of data provider

The workflow has also been implemented across HPC clusters, cloud-based virtual research environments, and public notebook platforms, confirming its portability across diverse computational ecosystems.

8.4 Implications for long-term uptake

The demonstrator shows that LiDAR-based vegetation structure metrics can be:

- Reproduced transparently using open-source tools
- Scaled from site-level studies to national monitoring programmes
- Deployed across diverse infrastructures and institutional settings
- Sustained through active software maintenance and versioned releases of core components (e.g. Laserchicken, Laserfarm, and NaaVRE)

Together, these characteristics provide a solid foundation for long-term integration within European research infrastructures and environmental monitoring frameworks beyond the lifetime of the MAMBO project. Continued software updates and documented release cycles ensure that the workflow can evolve in response to new methodological developments, computational technologies, and user requirements.

9 Contribution to MAMBO outcomes

The demonstrator directly supports the scientific and strategic objectives of MAMBO by operationalising scalable and reproducible workflows for LiDAR-based habitat condition assessment.

9.1 Methodological development

Within Work Package 4, the pipeline contributes by:

- Automating vegetation structure metric extraction from ALS data
- Standardising metric definitions across sites, countries, and time periods
- Demonstrating reproducible execution in distributed and cloud-based environments

These achievements confirm that harmonised LiDAR-derived structural metrics can be generated consistently across spatial scales.

9.2 Harmonisation and comparability

The workflow enhances comparability across regions through harmonised metric definitions and consistent processing logic. Explicit parameterisation of tiling schemes, spatial resolution, and classification rules ensures transparency and repeatability across heterogeneous datasets.

9.3 Large-scale monitoring capability

The SURF-based deployment demonstrates that:

- Multi-temporal national LiDAR archives (~70 TB uncompressed data) can be processed efficiently
- Vegetation structure metrics can be derived at country-wide scale using distributed computing

This provides a proof-of-concept for operational LiDAR-based monitoring at regional and national scales.

9.4 Accessibility and open science

The VRE and Colab prototypes broaden accessibility by supporting training and experimentation without requiring advanced HPC expertise. The use of open-source software, version-controlled repositories, and documented workflows aligns with FAIR and open science principles, ensuring transparency, reproducibility, and reuse.

9.5 Strategic impact

By validating the workflow across HPC, virtual research environments, and lightweight cloud platforms, the demonstrator establishes a scalable and transferable framework for LiDAR-based habitat condition monitoring. It positions the methodology for long-term uptake within European research infrastructures and national environmental monitoring programmes.

10 Conclusions and next steps

10.1 Conclusions

This deliverable presents a demonstrator pipeline for extracting vegetation structure metrics from airborne LiDAR using distributed and cloud-based computing environments. The workflow has been validated across three complementary implementations:

- HPC deployment on SURF for national-scale processing
- Containerised Virtual Research Environment (NaaVRE) prototype
- Lightweight Google Colab implementation

Together, these implementations demonstrate that the core workflow logic is portable, scalable, and reproducible across infrastructures. The pipeline enables harmonised vegetation structure metrics to be generated consistently across datasets, countries, and time periods, thereby supporting operational biodiversity monitoring at multiple scales.

10.2 Lessons learned

Key lessons include:

- Spatial tiling and distributed scheduling are fundamental for large-scale LiDAR processing
- Controlled software environments are essential for reproducibility
- Providing multiple access levels (HPC, VRE, Colab) broadens user adoption
- Infrastructure flexibility strengthens long-term sustainability

These insights provide guidance for future large-scale ecological remote sensing workflows.

10.3 Next steps

Future work may focus on:

- Integration into national and regional monitoring systems
- Application to additional European LiDAR archives
- Further automation and development of user-friendly configuration interfaces
- Integration with complementary remote sensing and in situ biodiversity data
- Continued training and capacity building through Virtual Research Environments and notebook-based platforms

10.4 Outlook

The demonstrator confirms that scalable LiDAR-based vegetation structure analysis is technically mature and operationally feasible. By validating the workflow across high-performance computing, virtual research environments, and lightweight cloud platforms, MAMBO establishes a robust foundation for the long-term adoption of harmonised, data-driven habitat condition monitoring in Europe.

11 Acknowledgements

The authors acknowledge France Gerard for feedback on an earlier draft of this document. We thank Meiert W. Grootes and Francesco Nattino from the Netherlands eScience Center (https://www.esciencecenter.nl/) for their continuous support in maintaining and further developing the Laserchicken and Laserfarm software.

We also acknowledge the NaaVRE team at the LifeWatch Virtual Laboratory & Innovation Centre (https://www.lifewatch.eu/organisation-governance/vlab-innovation-centre/), including Zhiming Zhao, Gabriel Pelouze, Spiros Koulouzis, Koen Greuell, and Quan Pan, for developing and supporting the NaaVRE LiDAR Virtual Lab.

This work was funded by the European Commission under the Horizon Europe Framework Programme (MAMBO project, grant no. 101060639). Additional support was provided by LifeWatch ERIC (https://www.lifewatch.eu/), a European Research Infrastructure Consortium focused on biodiversity and ecosystem research, and by SURF (https://www.surf.nl/en), the Dutch national provider of advanced ICT infrastructure for research and education.

12 References

Kissling, W. D., Mulder, W., Wang, J., & Shi, Y. (2025). Data of vegetation structure metrics retrieved from airborne laser scanning surveys for European demonstration sites. *Data in Brief*, 60, 111548. https://doi.org/10.1016/j.dib.2025.111548.

Kissling, W. D., Shi, Y., Koma, Z., Meijer, C., Ku, O., Nattino, F., et al. (2022). Laserfarm – A high-throughput workflow for generating geospatial data products of ecosystem structure from airborne laser scanning point clouds. *Ecological Informatics*, 72, 101836. https://doi.org/10.1016/j.ecoinf.2022.101836.

Kissling, W. D., Shi, Y., Koma, Z., Meijer, C., Ku, O., Nattino, F., et al. (2023). Country-wide data of ecosystem structure from the third Dutch airborne laser scanning survey. *Data in Brief*, 46, 108798. https://doi.org/10.1016/j.dib.2022.108798.

Kissling, W. D., Shi, Y., Wang, J., Walicka, A., George, C., Moeslund, J. E., et al. (2024). Towards consistently measuring and monitoring habitat condition with airborne laser scanning and unmanned aerial vehicles. *Ecological Indicators*, 169, 112970. https://doi.org/10.1016/j.ecolind.2024.112970.

Meijer, C., Grootes, M. W., Koma, Z., Dzigan, Y., Gonçalves, R., Andela, B., et al. (2020). Laserchicken—A tool for distributed feature calculation from massive LiDAR point cloud datasets. *SoftwareX*, 12, 100626. https://doi.org/10.1016/j.softx.2020.100626.

Shi, Y., Wang, J., & Kissling, W. D. (2025). Multi-temporal high-resolution data products of ecosystem structure derived from country-wide airborne laser scanning surveys of the Netherlands. *Earth System Science Data*, 17(7), 3641-3677. https://doi.org/10.5194/essd-17-3641-2025.

Wang, Y., Koulouzis, S., Bianchi, R., Li, N., Shi, Y., Timmermans, J., et al. (2022). Scaling notebooks as re-configurable cloud workflows. *Data Intelligence*, 4(2), 409-425. https://doi.org/10.1162/dint_a_00140.

Zhao, Z., Koulouzis, S., Bianchi, R., Farshidi, S., Shi, Z., Xin, R., et al. (2022). Notebook-as-a-VRE (NaaVRE): From private notebooks to a collaborative cloud virtual research environment. *Software: Practice and Experience*, 52(9), 1947-1966. https://doi.org/10.1002/spe.3098.

MAMBO
MODERN APPROACHES TO THE
MONITORING OF BIODIVERSITY

www.mambo-project.eu

Project partners

AARHUS UNIVERSITY
Naturalis Biodiversity Center
UFZ
University of Reading
Inria
PENSOFT
UKCEH
cirad
ECOSTACK INNOVATIONS LIMITED