py4dgeo: Software Structure and Workflow Concepts

Tutorial held at ISPRS Congress 2026

July 2026

Prof. Dr. **Bernhard Höfle**
Ronald Tabernig, MSc.

Prof. Dr. **Katharina Anders**
Xiaoyu Huang, MSc.

Core idea: Modular change analysis pipeline

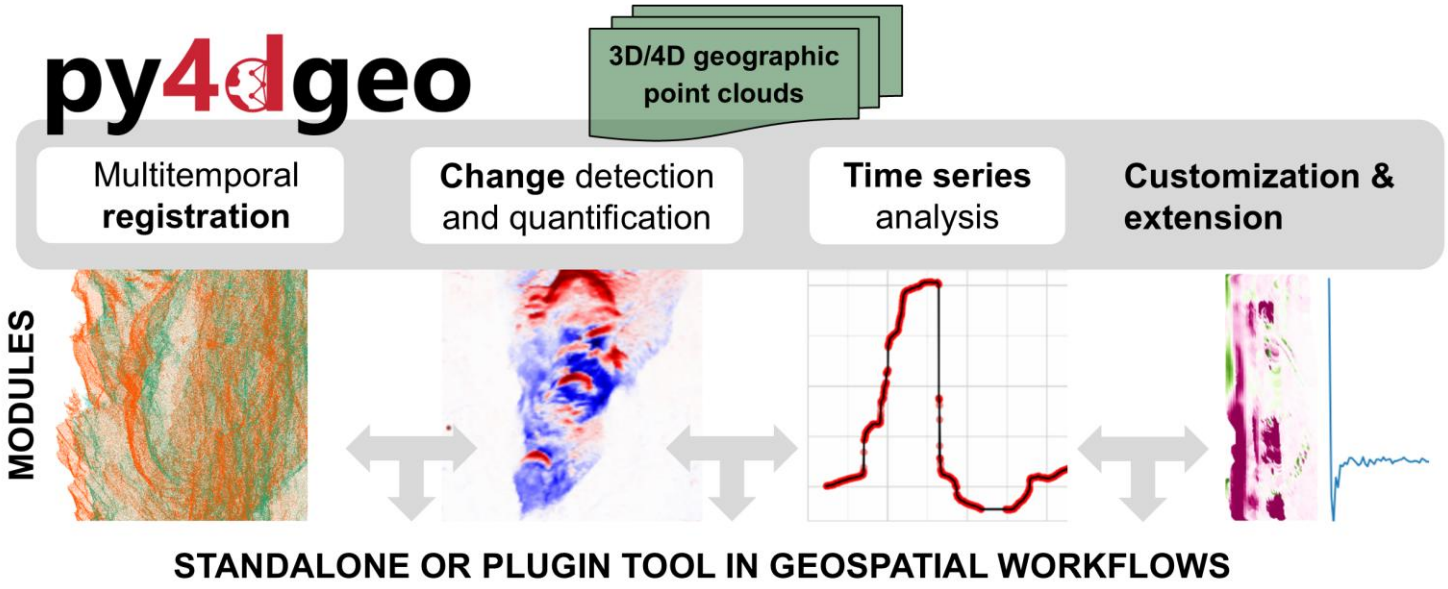


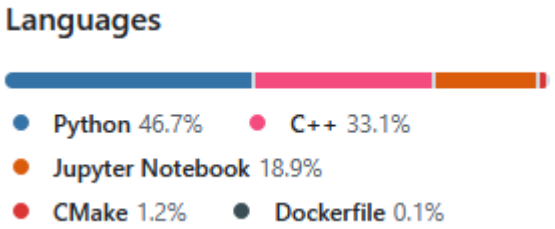
Figure from Anders et al. (2026)

- Full method portfolio in py4dgeo
- Modular → methods can be interchanged and combined
- Can be used standalone or in workflows

Architecture: Python + C++

Separation performance vs. scientific workflow design

- Hybrid design:
  - C++ core → performance-oriented
  - Python interface → flexibility
- pybind11 bindings
- CPU parallelized



Data model: core abstractions

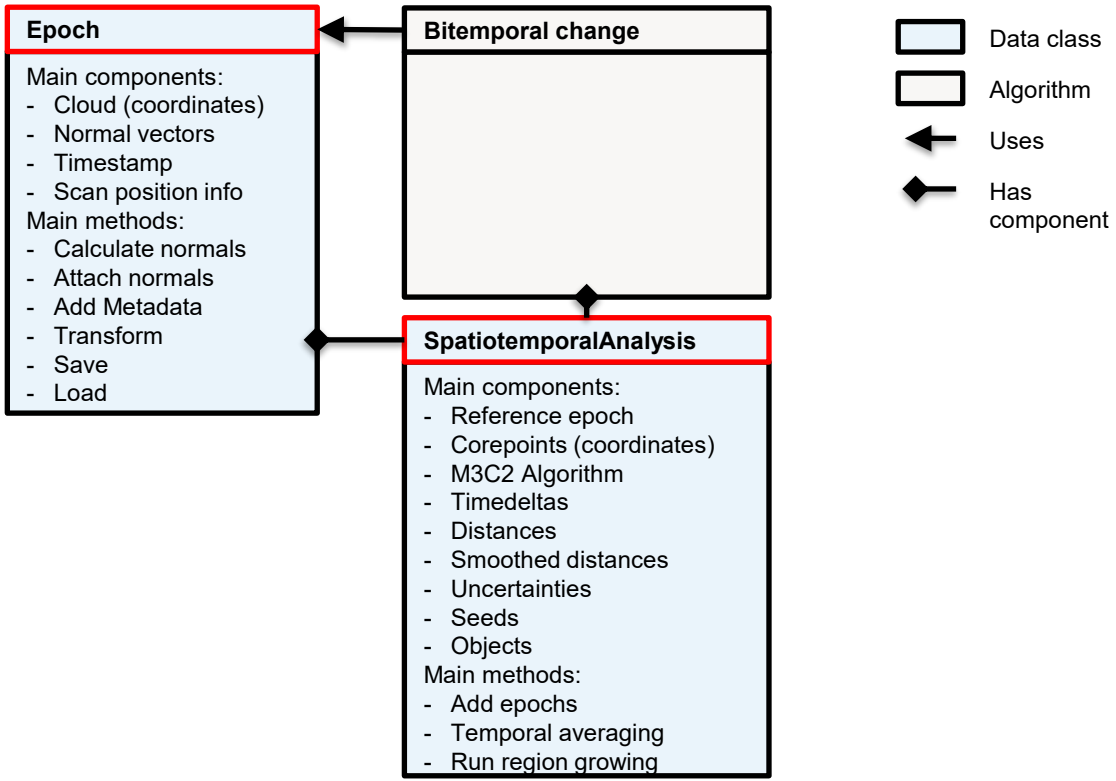


Figure modified from Anders et al. (2026)

SpatiotemporalAnalysis object

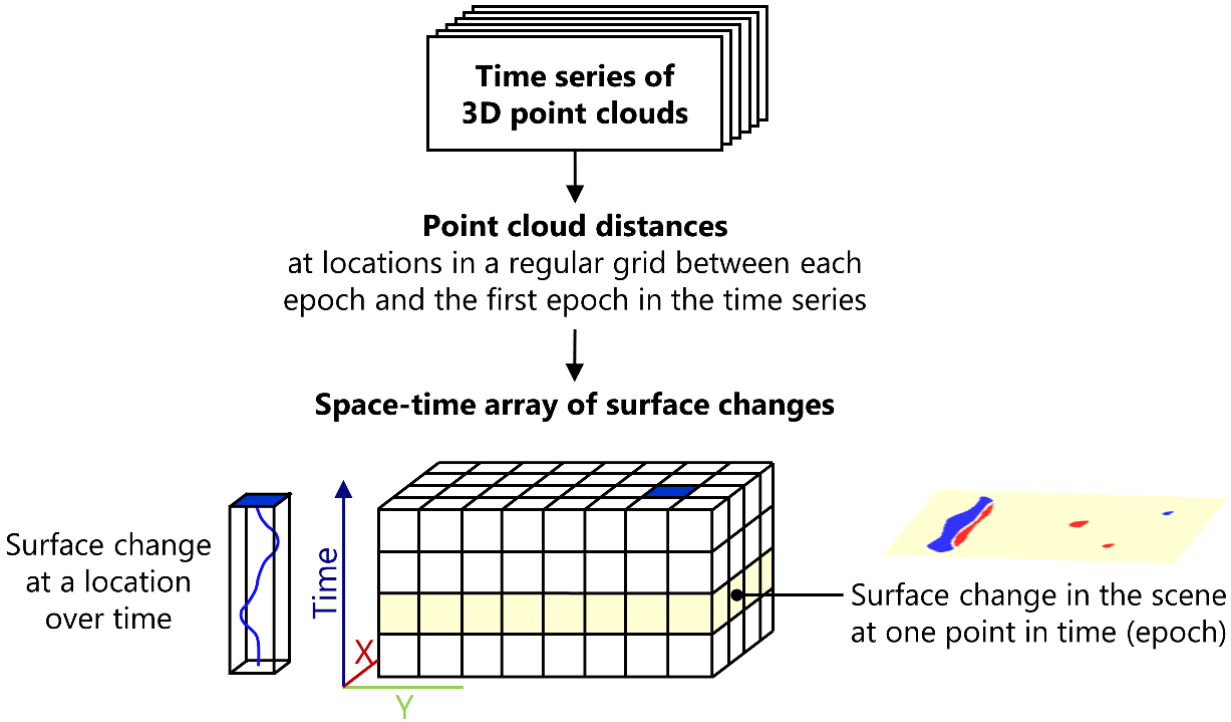


Figure by K. Anders modified from Anders et al. (2021)

SpatiotemporalAnalysis

Main components:
- Reference epoch
- Corepoints (coordinates)
- M3C2 Algorithm
- Timedeltas
- Distances
- Smoothed distances
- Uncertainties
- Seeds
- Objects

Main methods:
- Add epochs
- Temporal averaging
- Run region growing

Main method components

Epoch

Main components:
- Cloud (coordinates)
- Normal vectors
- Timestamp
- Scan position info

Main methods:
- Calculate normals
- Attach normals
- Add Metadata

M3C2 algorithm

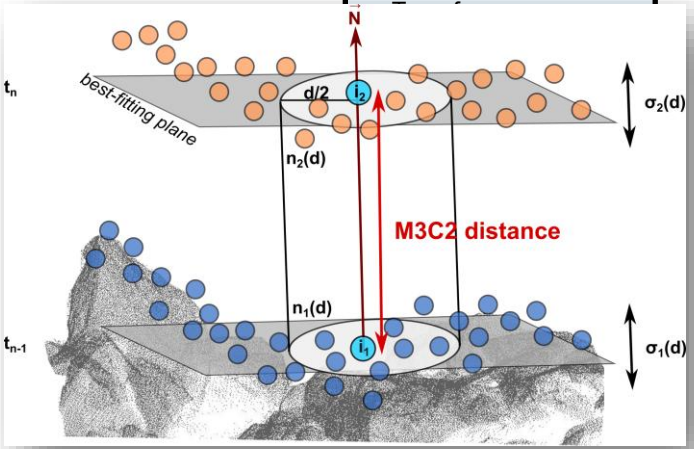
Main parameters:
- Epochs
- Corepoints (coordinates)
- Normal radii
- Cylinder radius
- Orientation vector
- Maximum distance
- Registration error

Data class

Algorithm

Uses

Has component



Schematic of point cloud distance computation with the M3C2 algorithm (Lague et al., 2013). Figure by V. Zahs modified from Zahs et al. (2022).

Figure modified from Anders et al. (2026)

MULTITEMPORAL **REGISTRATION** | **CHANGE** DETECTION AND QUANTIFICATION | **TIME SERIES**-BASED ANALYSIS

Main method components

Change detection and quantification: M3C2 + variants

Iterative closest point

Main parameters:
- Reference epoch
- Epoch
- Maximum iterations
- Tolerance
- Reduction point

ICP with stable areas

Main parameters:
- Reference epoch
- Epoch
- Initial distance threshold
- Level of detection
- Reference supervoxel resolution
- Supervoxel resolution
- Minimum number of supervoxel points
- Reduction point

Registration: ICP + stable area estimation

Epoch

Main components:
- Cloud (coordinates)
- Normal vectors
- Timestamp
- Scan position info

Main methods:
- Calculate normals
- Attach normals
- Add Metadata
- Transform
- Save
- Load

M3C2 algorithm

Main parameters:
- Epochs
- Corepoints (coordinates)
- Normal radii
- Cylinder radius
- Orientation vector
- Maximum distance
- Registration error

SpatiotemporalAnalysis

Main components:
- Reference epoch
- Corepoints (coordinates)
- M3C2 Algorithm
- Timedeltas
- Distances
- Smoothed distances
- Uncertainties
- Seeds
- Objects

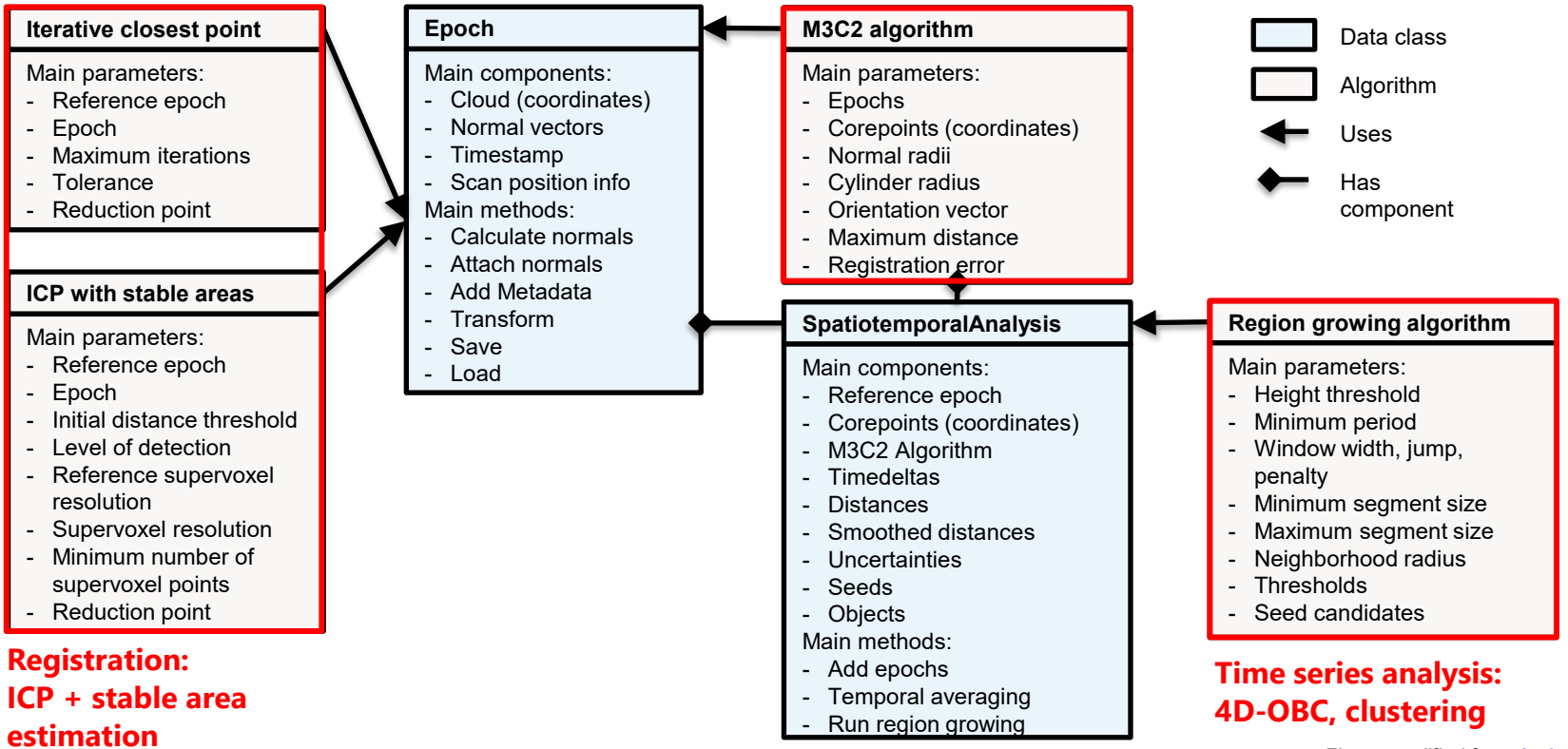
Main methods:
- Add epochs
- Temporal averaging
- Run region growing

Data class
Algorithm
Uses
Has component

Figure modified from Anders et al. (2026)

Main method components

Change detection and quantification: M3C2 + variants



Registration: ICP + stable area estimation

Time series analysis: 4D-OBC, clustering

Figure modified from Anders et al. (2026)

Minimal workflow example

1. Load data: create Epoch objects
2. Apply co-registration / fine alignment
3. Define analysis: reference epoch, core points, parameters
4. Run analysis: M3C2 / time series methods

```python
# import the library for M3C2 distance calculation
import py4dgeo

# load point clouds as epoch objects
epoch_2009, epoch_2017 = py4dgeo.read_from_las(
    las_data2009_aligned, las_data2017
)
```

```python
corepoints = epoch_2009.cloud[::]
```

```python
m3c2 = py4dgeo.M3C2(
    epochs=(epoch_2009, epoch_2017),
    corepoints=corepoints,
    normal_radii = [r for r in range(2, 8 + 1)]  # result: [2, 3, 4, 5, 6, 7, 8]
    cyl_radius=2.0,
    max_distance=(50.0),
    registration_error=(1.31)
)
```

Minimal workflow example

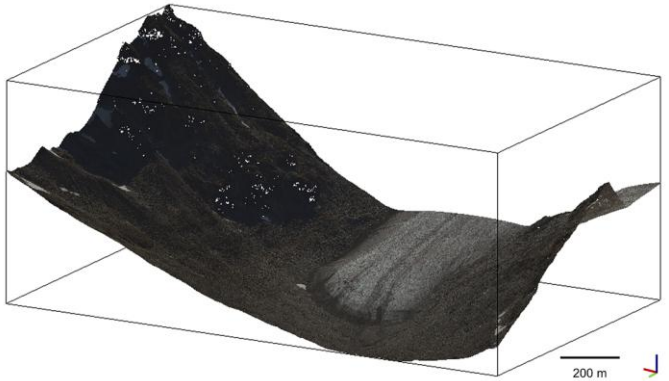
1. Load data: create Epoch objects
2. Apply co-registration / fine alignment
3. Define analysis: reference epoch, core points, parameters
4. Run analysis: M3C2 / time series methods

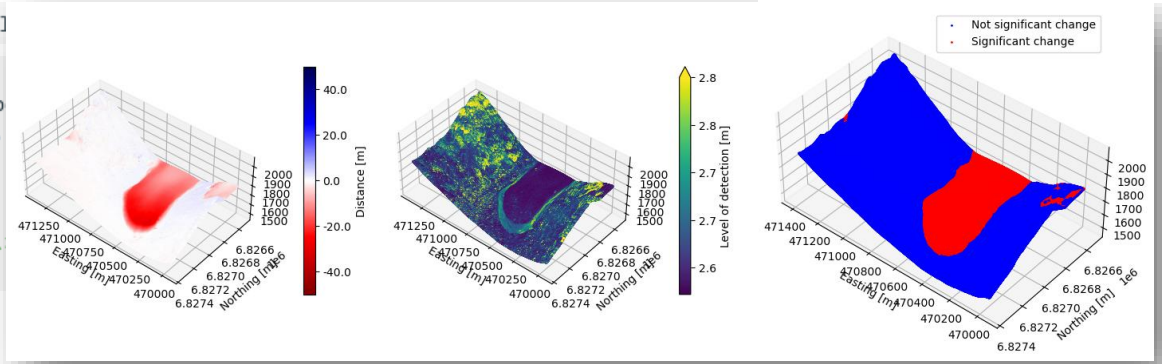
```python
# import the library for M3C2 distance calculation
import py4dgeo

# load point clouds as epoch objects
epoch_2009, epoch_2017 = py4dgeo.read_from_las(
    las_data2009_aligned, las_data2017
)
```

```python
corepoints = epoch_2009.cl
```

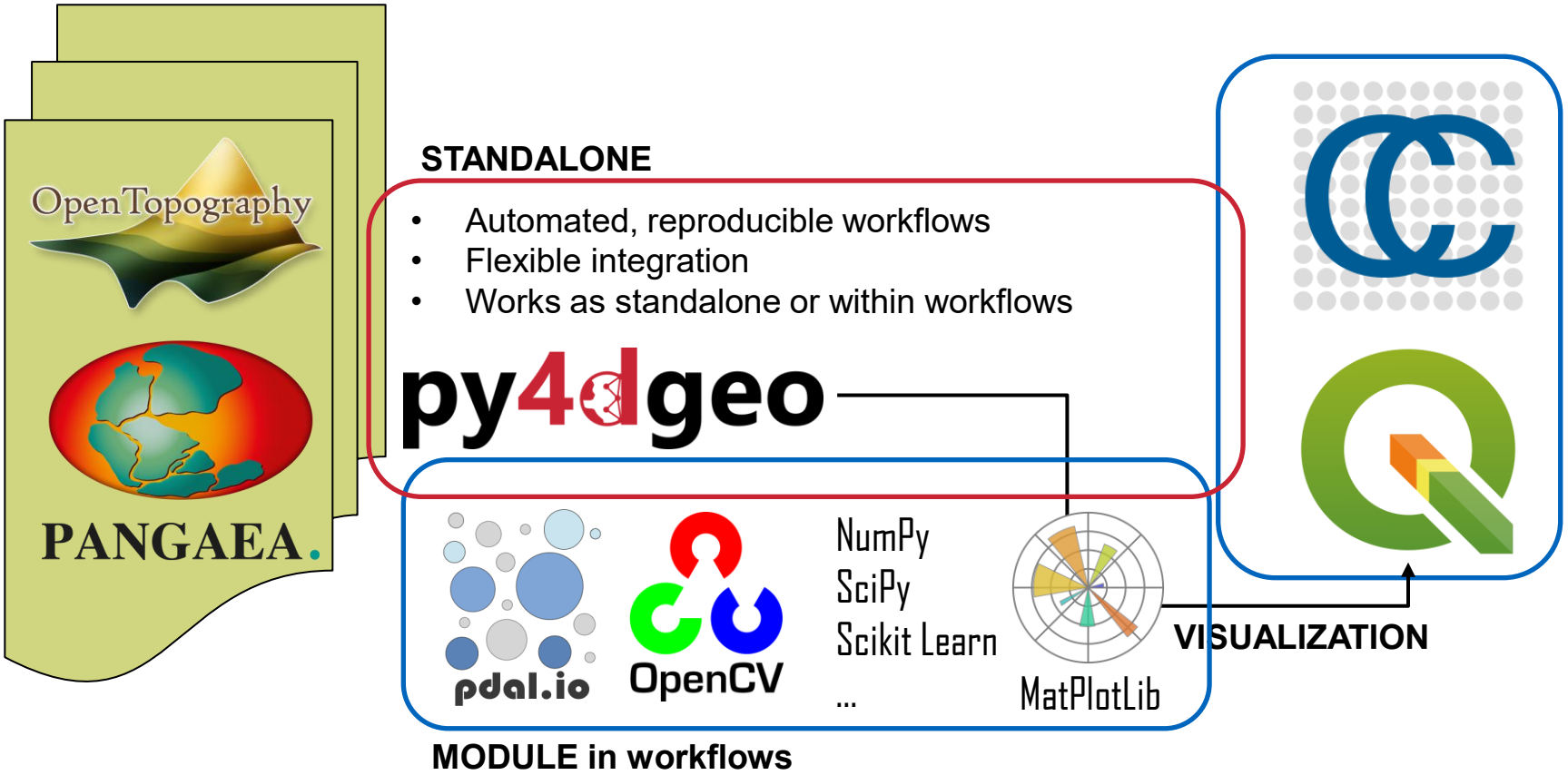
```python
m3c2 = py4dgeo.M3C2(
    epochs=(epoch_2009, ep
    corepoints=corepoints,
    normal_radii = [r for
    cyl_radius=2.0,
    max_distance=(50.0),
    registration_error=(1.
)
```





Figures by E-TRAINEE authors.
Data © Kartverket / CC BY 4.0.

Interoperability in the 3D/4D ecosystem



Customization and Extensibility

- Callbacks for algorithms
- Plug in your own methods
- Prototype in Python → optimize in C++

→ Framework for method development (beyond application)

Customization: Callbacks for algorithms

Each algorithm is represented by a class that inherits from **M3C2LikeAlgorithm** or from a more specialized class like **M3C2**.

Example: changing search directions (i.e. providing pre-defined normal vectors)

```python
class RenamedAlgorithm(py4dgeo.M3C2):
    @property
    def name(self):
        return "My super-duper M3C2 algorithm"
```

```python
class DirectionAlgorithm(RenamedAlgorithm):
    def directions(self):
        return np.array([0, 0, 1])
```

```
DirectionAlgorithm(epochs=(epoch1, epoch2), corepoints=corepoints, cyl_radius=5.0).run()
```

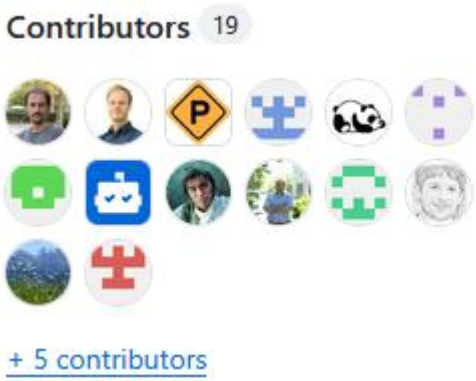
See: https://py4dgeo.readthedocs.io/en/latest/customization.html

For developers: Adding Python fallbacks to the C++ implementation
(https://py4dgeo.readthedocs.io/en/latest/customization.html#Adding-Python-callbacks-to-the-C++-implementation)

Reproducibility & community

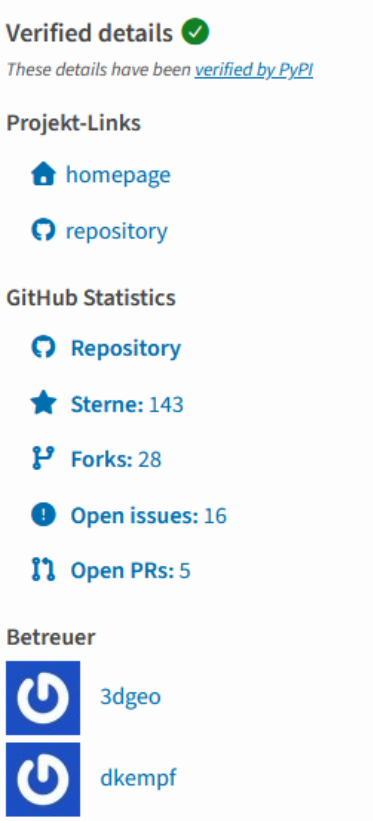
Built as a research software:

- Open source (MIT license)
- Jupyter demos with real datasets
- Documented workflows
- Continuous testing / CI
- Multi-institution development board



Some usage statistics
Downloads last day: 29
Downloads last week: 188
Downloads last month: 1,100

(https://pypistats.org/packages/py4dgeo, 2026-06-24)



Next in the agenda…

Questions and expectations?

Short break

Hands-on introduction
- Work with real datasets
- Run full workflows
- Modify parameters & methods