openEO: open science for Earth Observation compute platforms

Edzer Pebesma, Jun 26, 2024
OEMC Big Data Hackaton @ IIASA

Institute for Geoinformatics
University of Münster

What is open science?

It means that beyond sharing scientific papers (open access?) that explain what has been done, we also share how it was done, by:

- Sharing all data used, in a reusable form (FAIR)
- Sharing code, so the results (figures/tables) can be easily reproduced & reused, and findings can be scrutinized
- Using open source software, so that licenses are not obstructing reuse
- (making sure that versions are documented, if needed additionally share entire runtime, e.g. in form of docker images)

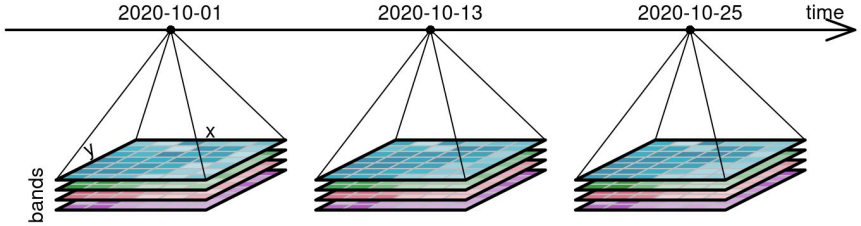
How does this work when the data used is too large to download?

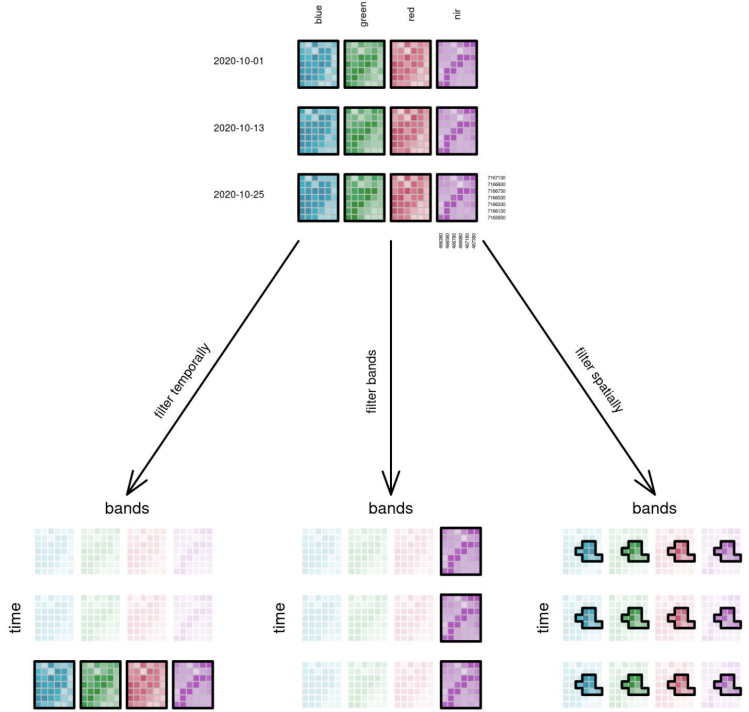
Cloud-based processing platforms

Suppose you carry out a land cover classification:

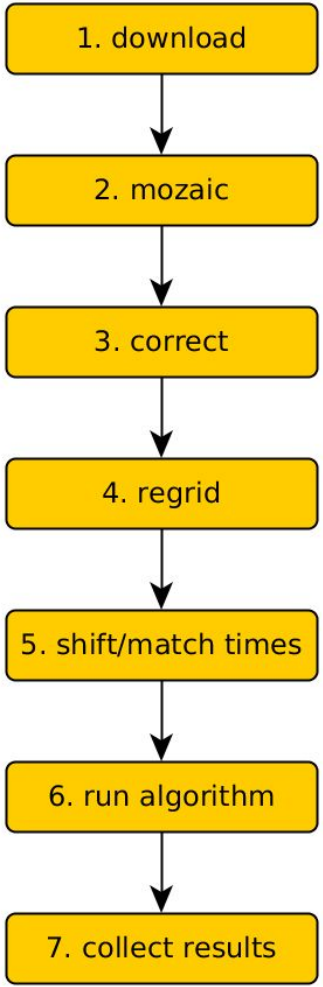
- using the same method and the same data, you do this on Google Earth Engine, and then on Sentinel-Hub. The results are largely similar, but the differences are too large to ignore.
- How do you continue?
  - How can you verify which of the results is the preferred one?
  - Where do you go to to ask what is going on, or what to do next?
- Note that both platforms (can) use a scripting interface, but the software actually doing computation (“back-end”) is closed source for both
- Who can compare the two scripts? Who is willing to do so?

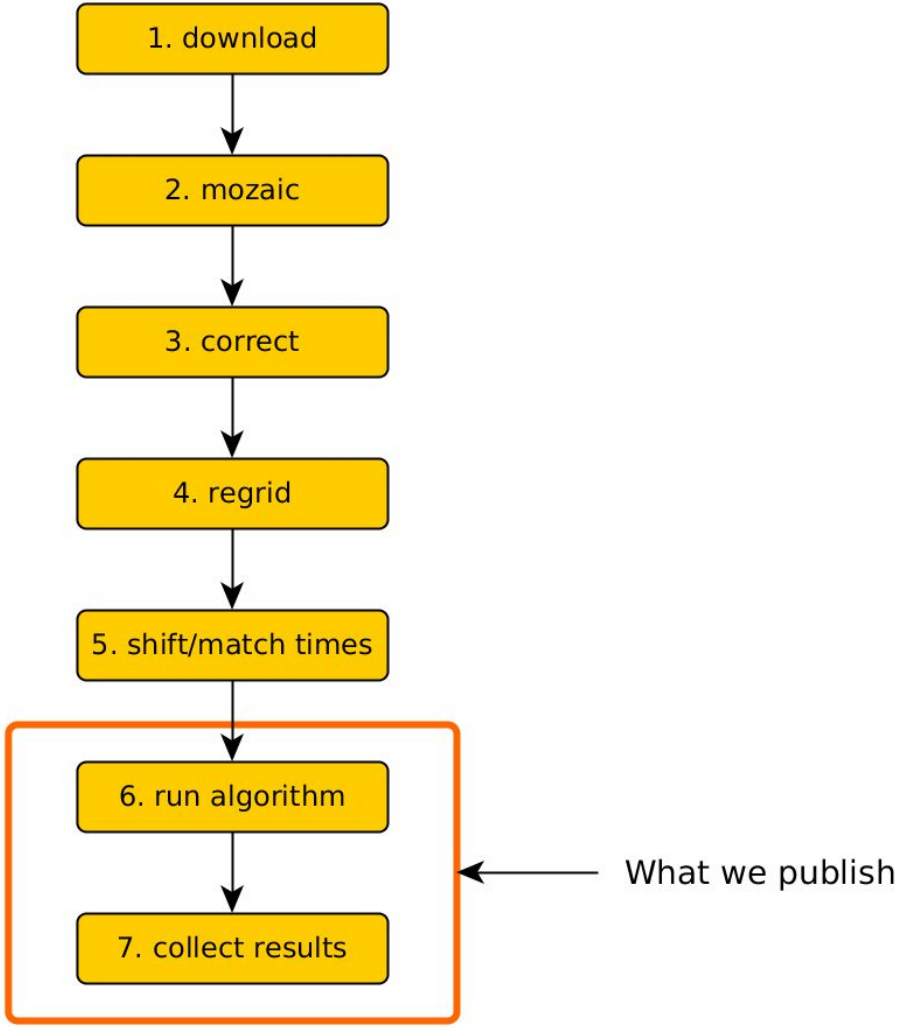
Data cubes:

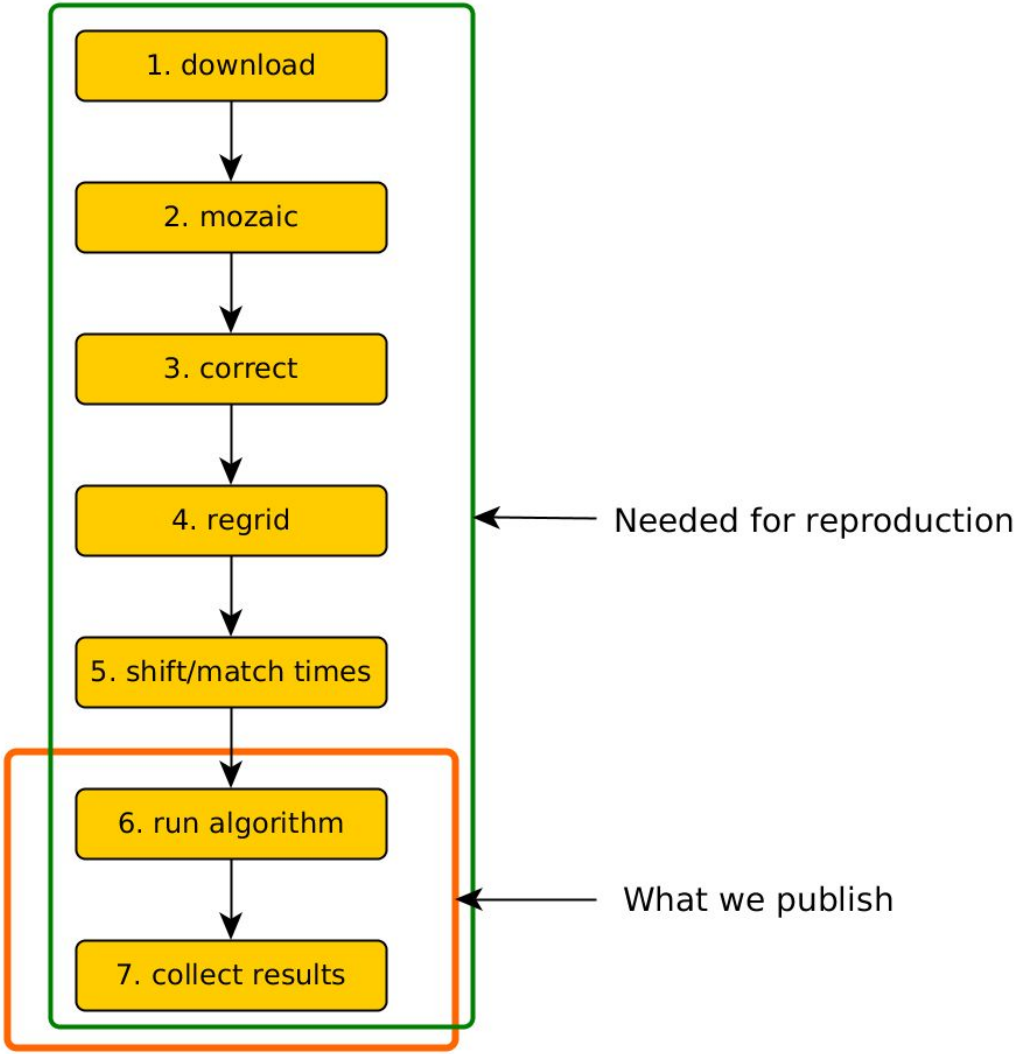


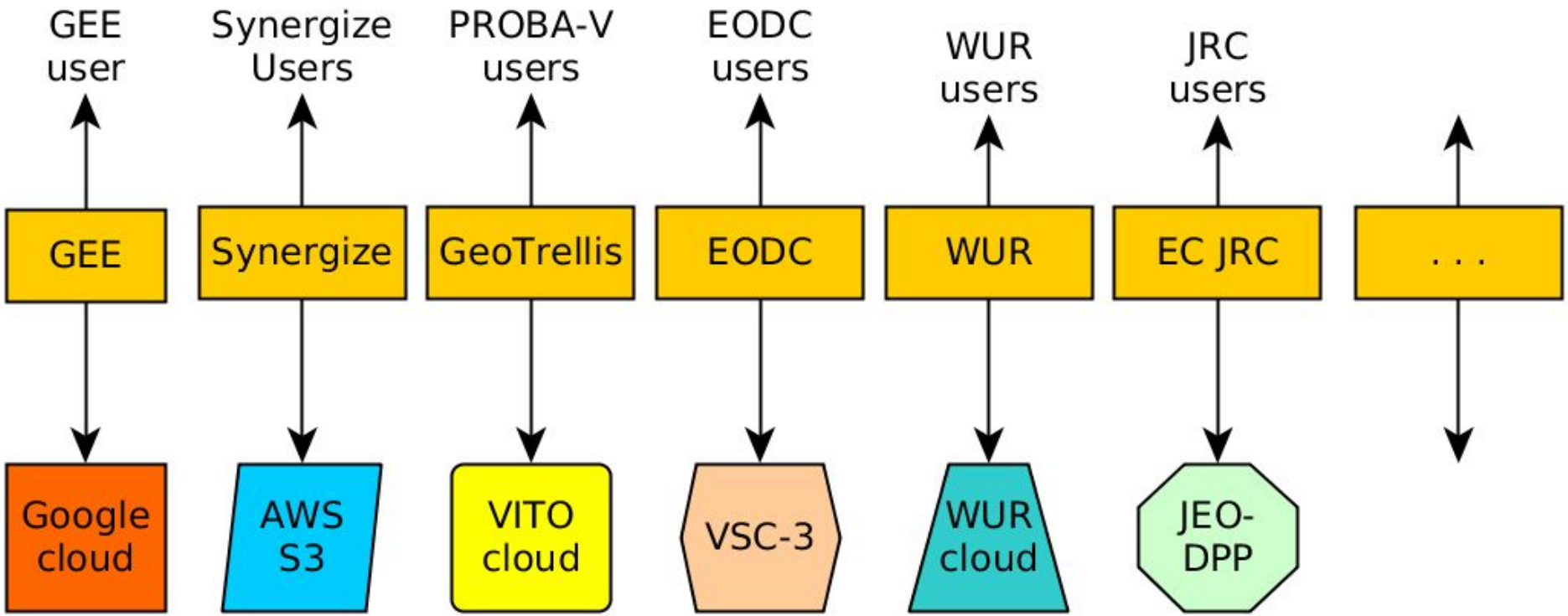


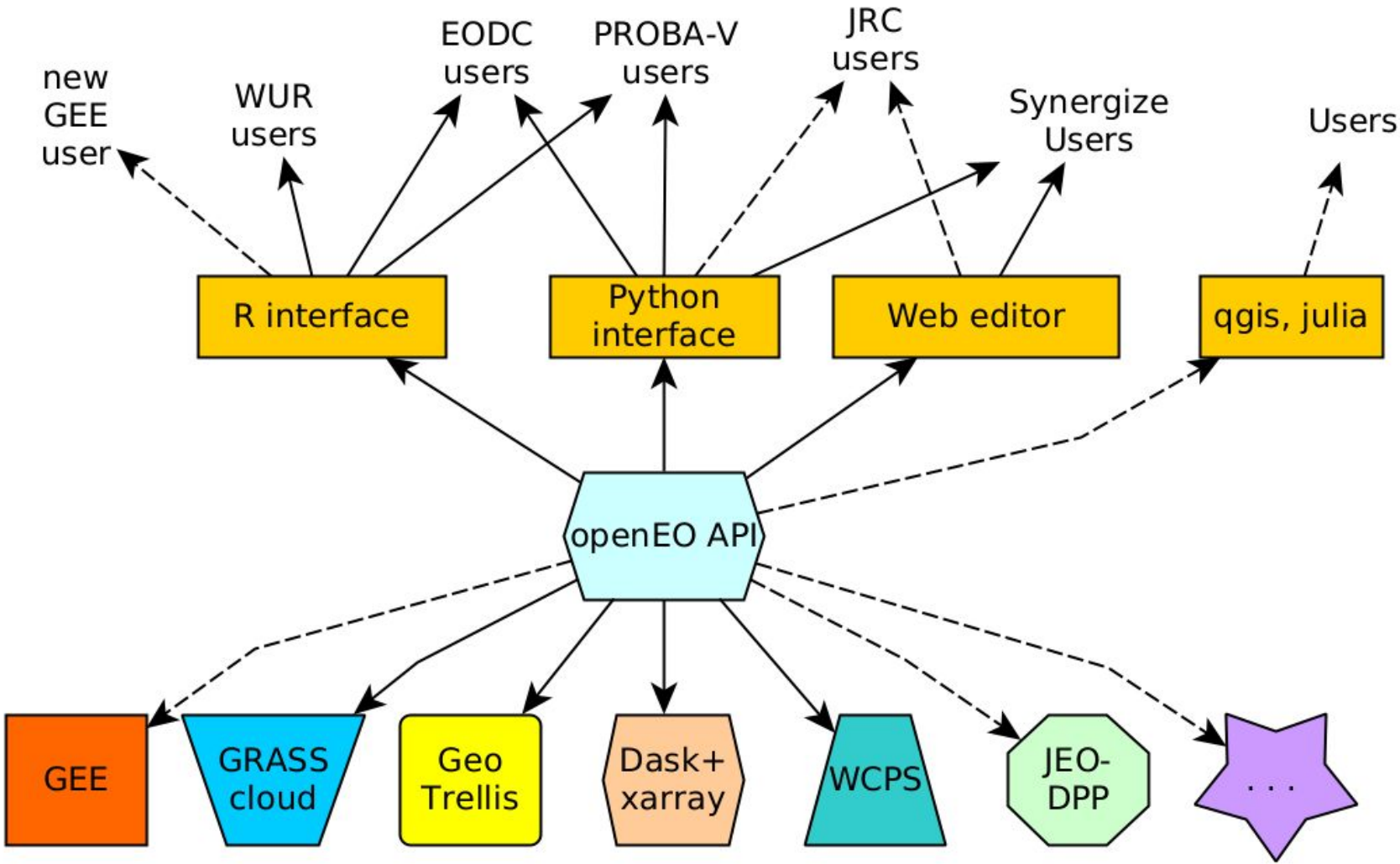
… also apply to model data (ERA5, CMIP6)

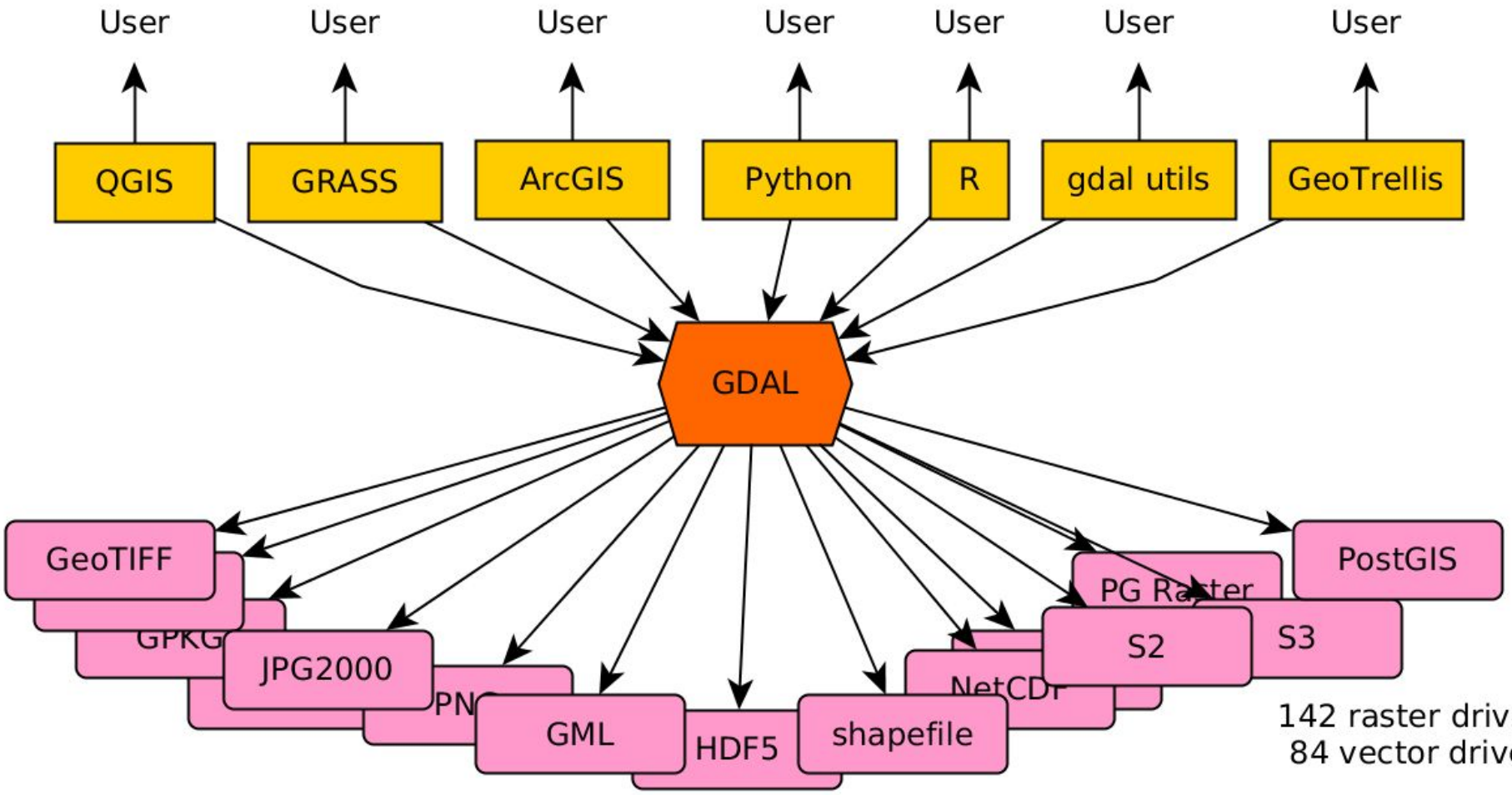

1. download
2. mozaic
3. correct
4. regrid
5. shift/match times
6. run algorithm
7. collect results

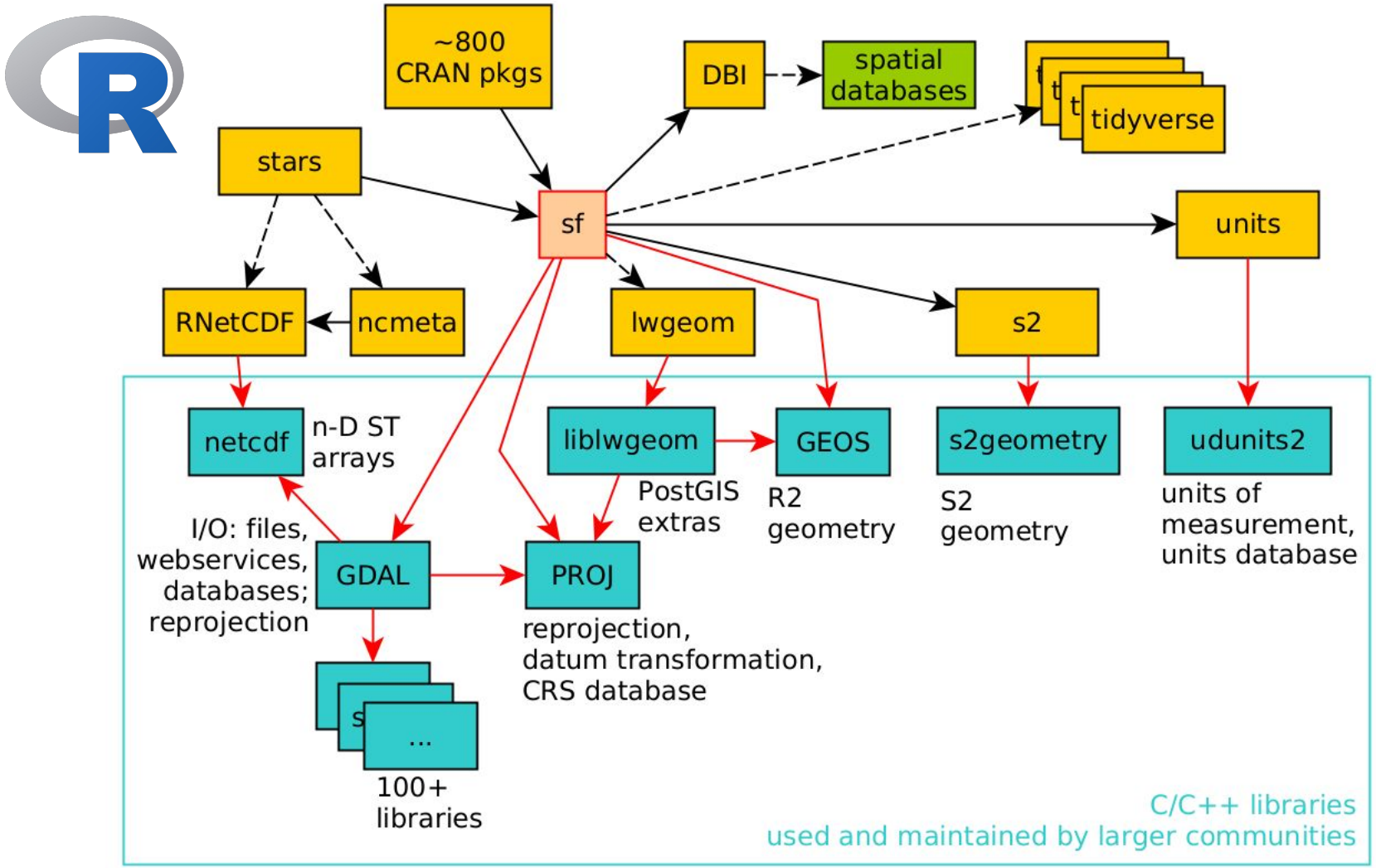

1. download
2. mozaic
3. correct
4. regrid
5. shift/match times
6. run algorithm
7. collect results
What we publish


1. download
2. mozaic
3. correct
4. regrid
5. shift/match times
6. run algorithm
7. collect results
Needed for reproduction
What we publish


GEE user
Synergize Users
PROBA-V users
EODC users
WUR users
JRC users
GEE
Synergize
GeoTrellis
EODC
WUR
EC JRC
. . .
Google cloud
AWS S3
VITO cloud
VSC-3
WUR cloud
JEO-DPP


new GEE user
WUR users
EODC users
PROBA-V users
JRC users
Synergize Users
Users
R interface
Python interface
Web editor
qgis, julia
openEO API
GEE
GRASS cloud
Geo Trellis
Dask+ xarray
WCPS
JEO-DPP
. . .


User
User
User
User
User
User
User
QGIS
GRASS
ArcGIS
Python
R
gdal utils
GeoTrellis
GDAL
GeoTIFF
GPKG
JPG2000
PN
GML
HDF5
shapefile
NetCDF
S2
S3
PG Raster
PostGIS
142 raster drivers
84 vector drivers


~800
CRAN pkgs
DBI
spatial
databases
tidyverse
stars
sf
units
RNetCDF
ncmeta
lwgeom
s2
netcdf
n-D ST
arrays
liblwgeom
GEOS
s2geometry
udunits2
PostGIS
extras
R2
geometry
S2
geometry
units of
measurement,
units database
I/O: files,
webservices,
databases;
reprojection
GDAL
PROJ
reprojection,
datum transformation,
CRS database
...
100+
libraries
C/C++ libraries
used and maintained by larger communities

What does, in the openEO context, “API” mean?

- API is a specification how computers interact
- For instance, APIs make the internet work
- openAPI is an industry standard for defining an API, which makes it for software developers easy to use (lots of tooling available)
- It abstracts problems beyond language silo’s like R, Python, Julia, Rust, …
- All data science languages can cope well with APIs, both client & server side
- openEO clients are available in R, Python, Julia, JavaScript
- A web editor uses the JavaScript client
- New clients are easy to develop
- New back-ends are (relatively) easy to develop

openEO API (1.2.0)

Download OpenAPI specification: Download

openEO Consortium: openeo.psc@uni-muenster.de | URL: https://openeo.org | License: Apache 2.0

The openEO API specification for interoperable cloud-based processing of large Earth observation datasets.

Conformance class: `https://api.openeo.org/1.2.0`

openEO Documentation

API Principles

Language

In the specification the key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119 and RFC 8174.

openEO processes (2.0.0-rc.1)

Search in processes

absolute

Absolute value

MATH

Description

`absolute(number|null x) : number|null`

Computes the absolute value of a real number `x`, which is the "unsigned" portion of x and often denoted as |x|.

The no-data value `null` is passed through and therefore gets propagated.

Parameters

x*

A number.

> Data type: **number, null**

Return Value

The computed absolute value.

> Data type: **number, null**
>
> Minimum value (inclusive): 0

Examples

How does an openEO session work?

1. Connect to a back-end
2. Authenticate (only needed on a public platform, and if you want to process)
3. Look up available collections (STAC)
4. Look up available processes
5. Define problem (select collection(s), spatial extent, temporal extent, bands)
6. Define processing task (like in any programming language)
7. (check billing?)
8. Submit job / download result / view result

Only step 8 triggers actual computation on the back-end

Who pays for all this?

- If you run on your own hardware managed cloud resources, you pay yourself
- If you run on CDSE or openEO Platform, ESA’s NoR may sponsor you
- On e.g. openEO Platform you can also buy credits
- On CDSE / CreoDIAS, Earth on Amazon, or GCS, the large open EO collections are available, you don’t pay for storage
- The billing process, in particular for federated cloud processing, may be challenging

Where to go with questions?

- To the instructor who stands next to you, or your colleagues
- To the user form on openEO Platform
- If software related: to the GitHub.com/Open-EO issues

Topics
More

Categories
openEO Platform
Uncategorized
openEO.org
Forum Rules
All categories

Tags
python
jupyterlab
r
authentication
other
All tags

openEO Platform ▸ subcategories ▸ tags ▸ Top Latest

Topic	Replies	Views	Activity
📌 How to join this Forum openEO Platform This forum allows users of openEO platform to give feedback, exchange ideas, or report issues. Anyone on the web can read the major Forum categories and their entries. To gain writing access to the forum please follow tw… read more	3	1.1k	Apr 2022
📌 Welcome to this Forum! openEO Platform A very warm welcome to anyone in this forum! This forum allows users of openEO platform to give feedback, exchange ideas, or report issues. All communication on the forum is public and freely available to anyone on the… read more	1	476	Dec 2021
Aggregate_spatial openEO Platform	4	44	1d
Oelab 403: forbidden openEO Platform jupyterlab	8	299	14d
Aggregate_spatial not preserving GeoJSON feature properties openEO Platform	1	48	14d
Showing warning of Need Credits openEO Platform python, jupyterlab, authentication, editor	2	67	23d
Quality filters for Sentinel5P CH4 data openEO Platform python, jupyterlab	3	73	23 May
Where to find L2H/L2F data generation with OpenEO Platform/EODC? openEO Platform	13	154	17 May

github.com/Open-EO

Open-EO

Type / to search

Overview | Repositories 65 | Projects 3 | Packages | Teams 6 | People 55 | Settings

openEO

openEO develops an open API to connect R, Python, JavaScript and other clients to EO cloud back-ends in a simple and unified way.

153 followers | http://openeo.org/ | @open_EO | openeo.psc@uni-muenster.de

Follow

Pinned

Customize pins

openeo-api Public
The openEO API specification
91 | 11

openeo-processes Public
Interoperable processes for openEO's big Earth observation cloud processing.
HTML | 49 | 14

openeo-python-client Public
Python client API for OpenEO
Python | 142 | 36

openeo-r-client Public
R client package for working with openEO backends
R | 60 | 16

openeo-js-client Public
JavaScript and TypeScript client for the openEO API.
JavaScript | 14 | 6

openeo.org Public
openeo.org landing page
Vue | 5 | 15

Repositories

Find a repository... | Type | Language | Sort | New

openeo-geopyspark-driver Public
OpenEO driver for GeoPySpark (Geotrellis)
Python | 25 | Apache-2.0 | 4 | 228 (3 issues need help) | 3 | Updated 10 minutes ago

openeo-python-driver Public
Common parts of a Python driver implementation for OpenEO
Python | 11 | Apache-2.0 | 5 | 70 (1 issue needs help) | 1 | Updated 1 hour ago

openeo-geopyspark-integrationtests Public

View as: Public

You are viewing the README and pinned repositories as a public user.

You can create a README file visible to anyone.

Get started with tasks that most successful organizations complete.

Discussions

Set up discussions to engage with your community!

Turn on discussions

People

View all

Invite someone

Top languages

Python | JavaScript | Vue | Java | Jupyter Notebook

Most used topics

Manage

Cubes & Clouds

Cloud Native Open Data Science for Earth Observation

View Course details

EO College

Not Enrolled

Login to Enroll

Free

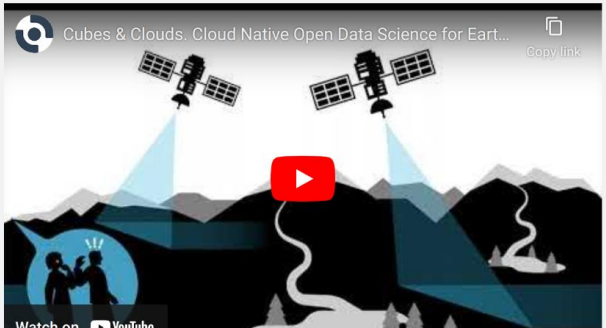
Course Includes

- 11 Lessons
- 30 Topics
- 10 Quizzes
- Course Certificate

What is this course about?

Welcome to **'Cubes & Clouds'**.

This course teaches the concepts of **data cubes**, **cloud platforms**, and **open science** in the context of earth observation.



openEO brings open science back to EO

… we managed to provide users a platform:

- Where they don’t have to worry about individual files, or managing cloud resources
- Which has support from ESA (openEO Platform) and the European Commission (CDSE)
- Which is built on multiple 100% open source back-ends, connects to existing closed source ones (GEE, Sentinel-Hub), and can be run locally
- Where analysis is based on data cubes, but integrates well with vector data and is in no way constrained to Earth Observation data (as PG showed yesterday)

… now consists of an active and lively ecosystem of users and developers

Many reasons to get involved!

https://zenodo.org/records/7254221:

ESA WorldCover 10 m 2021 v200

Zanaga, Daniele[1]; Van De Kerchove, Ruben[1]; Daems, Dirk[1]; De Keersmaecker, Wanda[1]; Brockmann, Carsten[2]; Kirches, Grit[2]; Wevers, Jan[2]; Cartus, Oliver[3]; Santoro, Maurizio[3]; Fritz, Steffen[4]; Lesiv, Myroslava[4]; Herold, Martin[5]; Tsendbazar, Nandin-Erdene[6]; Xu, Panpan[6]; Ramoino, Fabrizio[7]; Arino, Olivier[8]

Show affiliations

ESA WorldCover 10 m 2021 v200

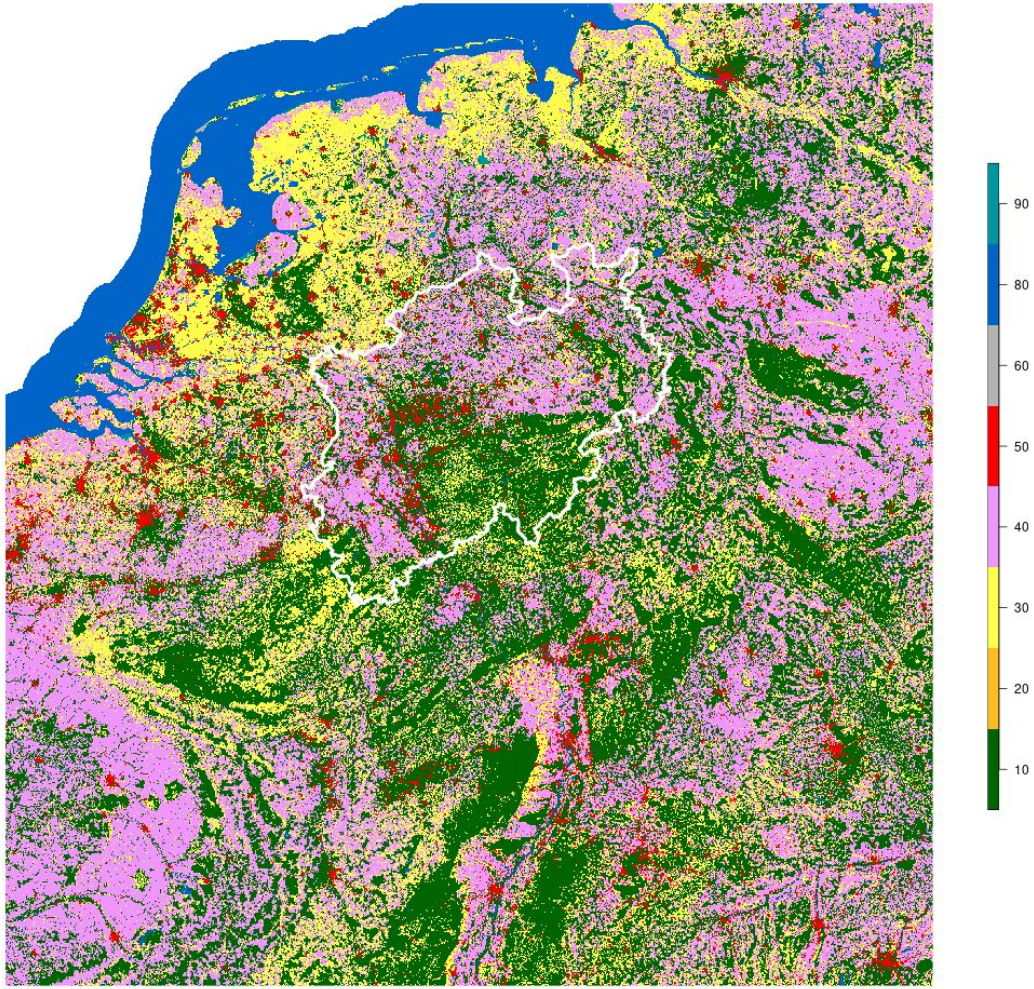
The European Space Agency (ESA) WorldCover 10 m 2021 product provides a global land cover map for 2021 at 10 m resolution based on Sentinel-1 and Sentinel-2 data. Th WorldCover product comes with 11 land cover classes, aligned with UN-FAO's Land Cover Classification System, and has been generated in the framework of the ESA WorldCover project.

The ESA WorldCover 10m 2021 v200 product updates the existing ESA WorldCover 10m 2020 v100 product to 2021 but is produced using an improved algorithm version (v200) compared to the 2020 map. Consequently, since the **WorldCover maps for 2020 and 2021 were generated with different algorithm versions** (v100 and v200, respectively), **changes between the maps** should be treated with caution, as they i**nclude both real changes in land cover and changes due to the algorithms used.**

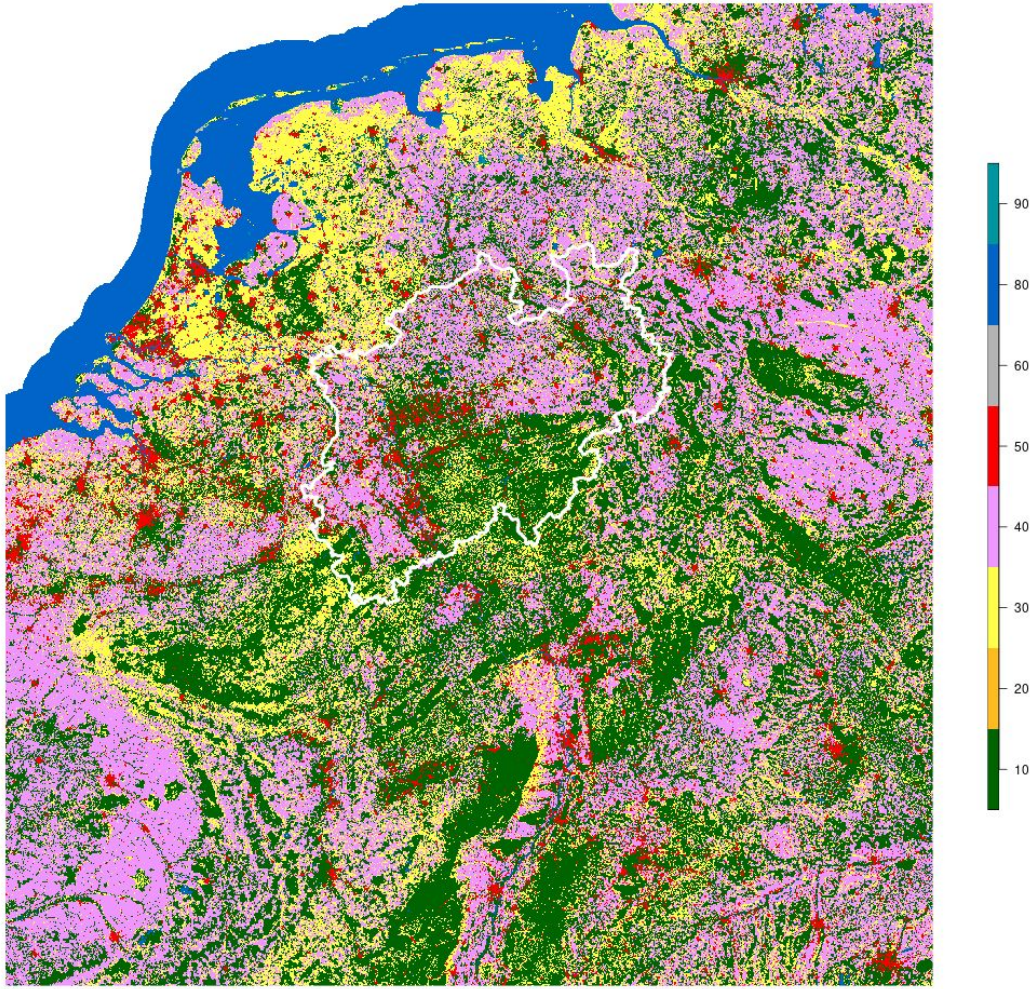
And in addition:

- Errors in the 2020 map
- Errors in the 2021 map
- Errors in both maps that do not cancel out

ESA wc 2020, v100



ESA wc 2021, V200



10 Tree cover

20 Shrubland

30 Grassland

40 Cropland

50 Built-up

60 Bare/sparse vegetation

70 Snow and ice

80 Permanent water bodies

90 Herbaceous wetland

95 Mangroves

100 Moss and lichen

Built-up:

2020: 6.36% +/- ?

2021: 8.03% +/- ?

Unbiased Area Estimation Using Copernicus High Resolution Layers and Reference Data

by Luca Kleinewillinghöfer [1,*], Pontus Olofsson [2], Edzer Pebesma [3], Hanna Meyer [4], Oliver Buck [1], Carsten Haub [1] and Beatrice Eiselt [5]

[1] EFTAS Fernerkundung Technologietransfer GmbH, Oststraße 2, 48145 Münster, Germany

[2] Department of Earth and Environment, Boston University Arts & Sciences, 685 Commonwealth Avenue, Boston, MA 02215, USA

[3] Institute for Geoinformatics, Westfälische Wilhelms-Universität Münster, Heisenbergstrasse 2, 48149 Münster, Germany

[4] Institute of Landscape Ecology, Westfälische Wilhelms-Universität Münster, Heisenbergstrasse 2, 48149 Münster, Germany

[5] European Commission, Eurostat, Rue Alphonse Weicker 5 L, 2721 Kirchberg, Luxembourg

* Author to whom correspondence should be addressed.

Remote Sens. **2022**, *14*(19), 4903; https://doi.org/10.3390/rs14194903

Submission received: 8 August 2022 / Revised: 23 September 2022 / Accepted: 26 September 2022 / Published: 30 September 2022

Abstract

Land cover area estimates can be derived via design-based approaches using a probability (random) reference sample. The collection of samples is usually costly and requires an effective sampling design. Earth-Observation-based mapping approaches do not have this requirement but can be biased in providing area estimates. Combining reference samples with remote sensing products can reduce sampling efforts and provide a more effective method to estimate land cover. The Copernicus High-Resolution Layer (HRL) provides remote-sensing-based data across Europe to support area estimation. Different methods are tested to estimate areas