Reproducible science, cloud-based Earth observation data processing, openEO

ifgi
Institute for Geoinformatics
University of Münster

openEO

Edzer Pebesma

GEO Expert Advisory Group Meeting, Sep 6-7, 2018

What does *I am a data scientist* mean?

I:

- have some domain knowledge (environmental, hydrology, RS)
- know some about spatial statistics
- communicate code

What does *I communicate code* mean?

I:

- write code (quite a lot)
- share code, mostly as R packages, but also in blogs and tweets
- engage in discussions concerning code (GH issues, SO, lists)
- actively engage with user and developer communities

I have a (meta-)scientific interest in what *reproducible geosciences* means.

What does *I communicate code* mean?

I:

- write code (quite a lot)
- share code, mostly as R packages, but also in blogs and tweets
- engage in discussions concerning code (GH issues, SO, lists)
- actively engage with user and developer communities

I have a (meta-)scientific interest in what *reproducible geosciences* means.

How does data science work?

To get things done, data scientists use languages, e.g. python, R, javascript or Julia, ... that share the following properties:

- they run everywhere, rather easily
- you can express problems in compact, human-readable and reproducible scripts
- unnecessary technical details (file format details, http calls, json/xml) are hidden
- code condensation is possible thanks to a package system, which also runs everywhere
- the language and package system are open source
- package systems resemble ecosystems
- analyses are shared/reproduced through Jupyter notebooks or R-markdown files.

What is reproducibility?

Here: if you give me your data and software, can I repeat your computations and get identical outcomes?

Why is this good?

- We want: *decisions and actions that are informed by Earth observation information and services* (abridged GEO vision)
- Information is derived from data through *analysis*, which involves computation
- Trust in information comes from a shared understanding of what the data are, how we analyse, and which computations were involved
- Computations happen with software
- When do we trust software?
  1. when a lot of people use it,
  2. when we can verify it by running tests, comparisons and benchmarks,
  3. when we can verify (and potentially improve) the source code.

What is reproducibility?

Here: if you give me your data and software, can I repeat your computations and get identical outcomes?
Why is this good?

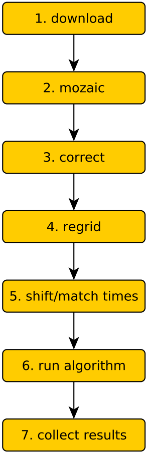
- We want: *decisions and actions that are informed by Earth observation information and services* (abridged GEO vision)
- Information is derived from data through *analysis*, which involves computation
- Trust in information comes from a shared understanding of what the data are, how we analyse, and which computations were involved
- Computations happen with software
- When do we trust software?
  1. when a lot of people use it,
  2. when we can verify it by running tests, comparisons and benchmarks,
  3. when we can verify (and potentially improve) the source code.

What is reproducibility?

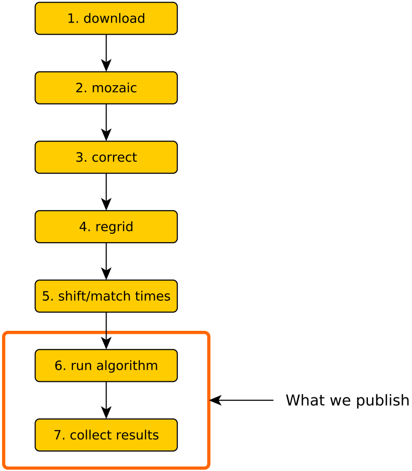
Here: if you give me your data and software, can I repeat your computations and get identical outcomes?
Why is this good?

- We want: *decisions and actions that are informed by Earth observation information and services* (abridged GEO vision)
- Information is derived from data through *analysis*, which involves computation
- Trust in information comes from a shared understanding of what the data are, how we analyse, and which computations were involved
- Computations happen with software
- When do we trust software?
  1. when a lot of people use it,
  2. when we can verify it by running tests, comparisons and benchmarks,
  3. when we can verify (and potentially improve) the source code.

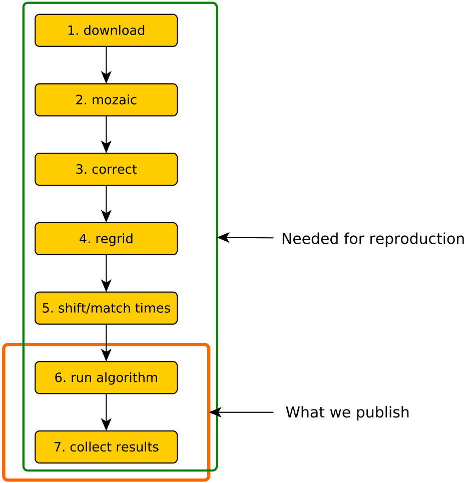
Current Earth Observation Research:



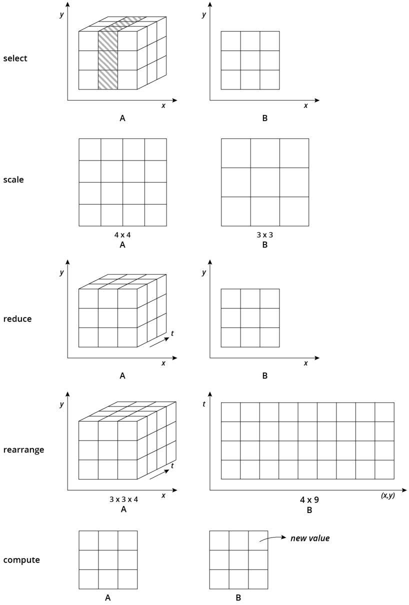
Current Earth Observation Research:



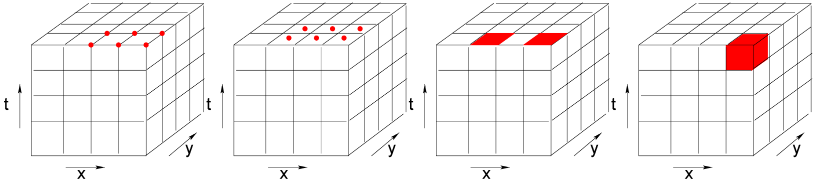
Current Earth Observation Research:



How do we *name* array operations



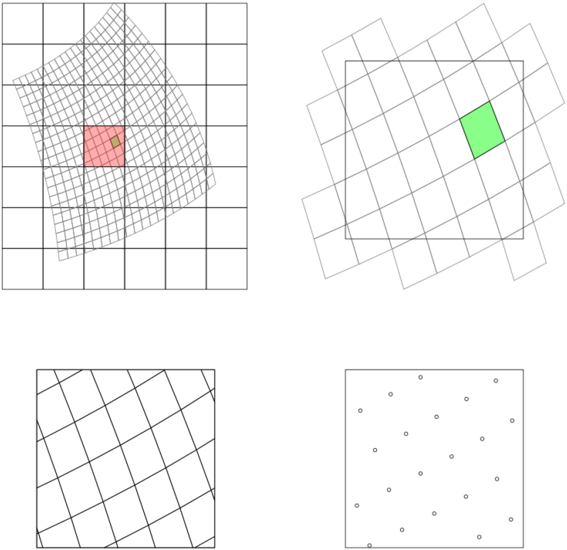
Is a pixel a point or an area?



M. Lu et al, Multidimensional Arrays for Analysing Geoscientific Data ISPRS Int. J. Geo-Inf. 2018, 7(8), 313;

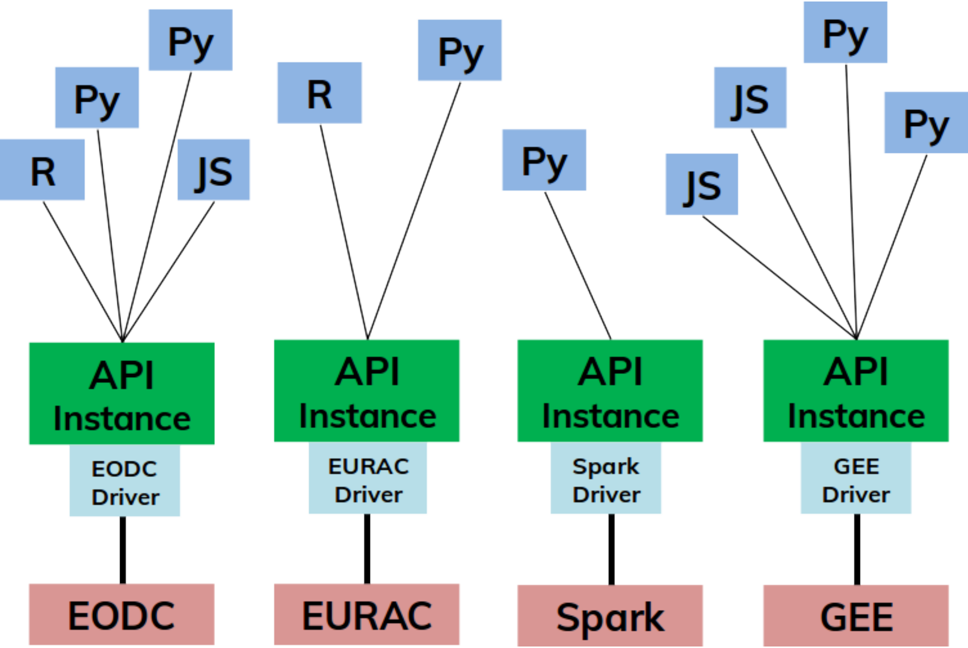
https://doi.org/10.3390/ijgi7080313

Is a pixel a point or an area?



openEO

- H2020, Oct 2017-2020,
- http://openeo.org/
- openEO develops an open API to connect R, python and javascript clients to big Earth observation cloud back-ends in a simple and unified way.



R
Py
Py
JS

API Instance

EODC Driver

EODC

R
Py

API Instance

EURAC Driver

EURAC

Py

API Instance

Spark Driver

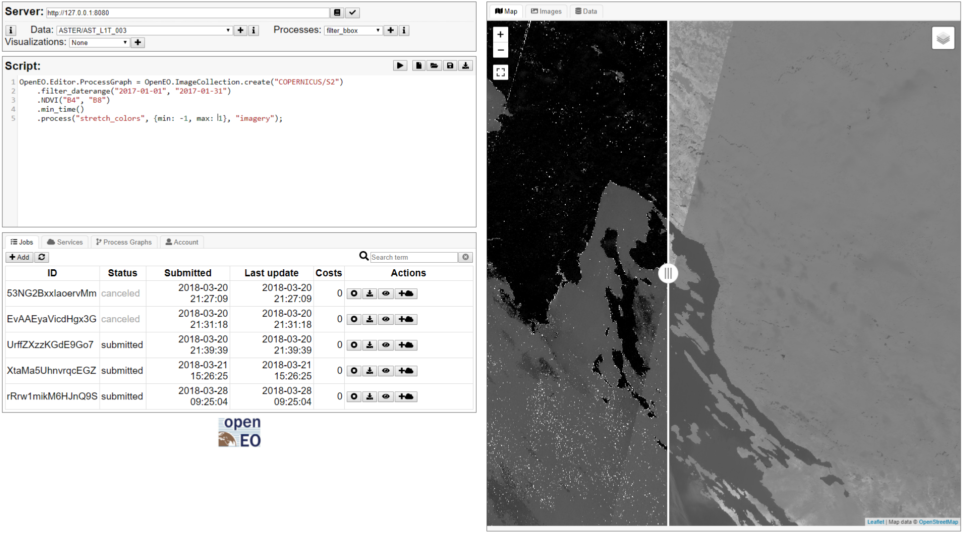
Spark

JS
JS
Py
Py

API Instance

GEE Driver

GEE



Server: http://127.0.0.1:8080

Data: ASTER/AST_L1T_003

Processes: filter_bbox

Visualizations: None

Script:

```
OpenEO.Editor.ProcessGraph = OpenEO.ImageCollection.create("COPERNICUS/S2")
    .filter_daterange("2017-01-01", "2017-01-31")
    .NDVI("B4", "B8")
    .min_time()
    .process("stretch_colors", {min: -1, max: 1}, "imagery");
```

ID	Status	Submitted	Last update	Costs	Actions
53NG2BxxlaoervMm	canceled	2018-03-20 21:27:09	2018-03-20 21:27:09	0	
EvAAEyaVicdHgx3G	canceled	2018-03-20 21:31:18	2018-03-20 21:31:18	0	
UrffZXzzKGdE9Go7	submitted	2018-03-20 21:39:39	2018-03-20 21:39:39	0	
XtaMa5UhnvrqcEGZ	submitted	2018-03-21 15:26:25	2018-03-21 15:26:25	0	
rRrw1mikM6HJnQ9S	submitted	2018-03-28 09:25:04	2018-03-28 09:25:04	0	

```
{ "process_id":"min_time",
  "args":{
    "imagery":{
      "process_id":"/user/custom_ndvi",
      "args":{
        "imagery":{
          "process_id":"filter_daterange",
          "args":{
            "imagery":{
              "process_id":"filter_bbox",
              "args":{
                "imagery":{
                  "product_id":"S2_L2A_T32TPS_20M"
                },
                "left":652000,
                "right":672000,
                "top":5161000,
                "bottom":5181000,
                "srs":"EPSG:32632"
              }
            },
            "from":"2017-01-01",
            "to":"2017-01-31"
          }
        },
        "red":"B04",
        "nir":"B8A"
      }
    }
  }
}
```

Cube view

File-agnostic access to EO imagery through a data cube view boosts usability of EO data.

In openEO:

- spatial dimensions are complemented with other dimensions such as the temporal or spectral dimensions
- researchers can directly filter, aggregate, or map functions over dimensions of a user-defined cube without being concerned about how the data in the processing platform is organised (granules, collections, coverages, ...)
- raster and vector data cubes are integrated.

Proof of Concept

The Month 6 (April 2018) proof of concept involved:

- coupling 3 clients (Python, R, JavaScript web-editor: figure left) to 7 back-ends (Sentinel Hub, GRASS GIS, EODC OpenStack, WCPS, Python GeoPySpark / GeoTrellis, Google Earth Engine, R) for
- 3 use-cases with band indexes, time series, aggregation over polygons, and user-defined (Python) functions
- source code and API docs on GitHub
- P.o.C. demo videos on the project web site

Why don't we build upon existing geo-standards?

- which standards?
- ...
- we do use non-geo standards where appropriate:
  - OpenAPI / swagger 3.0, allowing auto code generation for new clients
  - OAuth2 for user authentication
- no useful standards exist for describing (discovering, processing, publishing) image collection / dataset series;
- together with the GEE team, we do engage in describing these (in the space-time asset catalogue, STAC)

Why don't we build upon existing geo-standards?

- which standards?
- ...
- we do use non-geo standards where appropriate:
  - OpenAPI / swagger 3.0, allowing auto code generation for new clients
  - OAuth2 for user authentication
- no useful standards exist for describing (discovering, processing, publishing) image collection / dataset series;
- together with the GEE team, we do engage in describing these (in the space-time asset catalogue, STAC)

Upcoming challenges

- MANY!!
- A big one: UDFs (user-defined functions): how can I have my back-end execute my arbitrary (python, R) function on selected imagery?
- validating (verifying) back-ends against each other
- combining several back-ends
- User adoption: how/when will users start to adopt this (clients AND servers need to work, be useable, and be affordable!)

Towards a *results-oriented GEO*?

- provide (develop, share access to) compute nodes where big EO data can be *sensibly* analysed
- sensibly:
  - not by tile, but by image collection
  - providing a user-defined cube view
- implement simple methods (open source, or otherwise open API) to compute there
- develop benchmarks showing that these compute nodes produce the same results given the same queries (openEO)
- develop best practice executable documents (e.g. Jupyter notebooks) showing how one could use this software/service infrastructure to answer questions (openEO)
- create a sustainable ecosystem of users
- ... by creating a "pit of success"?