The Thuringian Data Cube

Implementation of an Earth Observation Data Cube for the Free State of Thuringia

Marco Wolsza

A thesis presented for the degree of
Master of Science

Supervised by:
Dr.-Ing. Clémence Dubois
Dr. Marcel Urban

Friedrich Schiller University Jena
June 18, 2021

I, Marco Wolsza confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

The amount of Earth Observation (EO) data being available through free and open data policies is continuously increasing and facilitates research that is essential to better understand the dynamics of the Earth's surface. It remains a challenge, however, to utilize EO data to its full potential and consistently derive valuable information, due to its complexity and growing volume. The novel concepts of Earth Observation Data Cubes (EODC) and Analysis Ready Data (ARD) are driving the development of innovative tools that enable data-intensive research in the EO domain, while easing the burden on users in terms of data processing and management.

The Thuringian Data Cube (TDC), an EODC based on the Open Data Cube software library, was implemented on a High Performance Computing system for the Free State of Thuringia in Germany. The initial implementation includes analysis-ready EO data from Landsat 8, Sentinel-2A/B and Sentinel-1A/B for a three-year period (2017–2019) and was facilitated by developing a software tool called ARDCube. Two use cases were conducted to demonstrate the potential of the TDC and assess its usability, including a time-series analysis investigating possible drought-related impacts on a forest area in eastern Thuringia.

The study demonstrates the capabilities of the TDC, and that it successfully lays a foundation to efficiently manage and analyze large volumes of EO data for the Free State of Thuringia. Various opportunities were identified to further improve the TDC in regard to the technical implementation and the integrated ARD products, as well as for subsequent analyses related to the use cases to be performed.

Acknowledgements

First of all, I would like to thank Dr.-Ing. Clémence Dubois and Dr. Marcel Urban for supervising this thesis and always providing valuable feedback. Beyond that, you always contributed to this being a rather pleasant experience, which I definitely did not take for granted!

Thank you John Truckenbrodt, for all the advice and interesting discussions. I'm looking forward to further explore the topics of this thesis with you and to also continue to learn from you in the process!

Thanks Markus and Antje for the valuable reviews you provided! And Markus also for joining many lunch breaks that often included discussions about relevant topics.

The realization of this project would not have been possible in this form without many awesome people who dedicate their time to contribute to open-source software projects. Thank you!

Finally, I'm of course deeply grateful for the continuous support and love from my parents and my sister. I would not have reached this far without you!

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AGDC	**A**ustralian **G**eoscience **D**ata **C**ube
AMA	**A**nalytical **M**echanics **A**ssociates
AOD	**A**tmospheric **O**ptical **D**epth
API	**A**pplication **P**rogramming **I**nterface
ARCO	**A**nalysis **R**eady **C**loud **O**ptimized
ARD	**A**nalysis **R**eady **D**ata
ARDC	**A**frican **R**egional **D**ata **C**ube
BDC	**B**razil **D**ata **C**ube
BOA	**B**ottom **O**f **A**tmosphere
BRDF	**B**idirectional **R**eflectance **D**istribution **F**unction
BSQ	**B**and **Sq**uential Interleaving
BT	**B**rightness **T**emperature
CARD4L	**C**EOS **A**nalysis **R**eady **D**ata **for L**and
CEOS	**C**ommittee on **E**arth **O**bservation **S**atellites
COG	**C**loud **O**ptimized **G**eoTIFF
CRS	**C**oordinate **R**eference **S**ystem
CSIRO	**C**ommonwealth **S**cientific and **I**ndustrial **R**esearch **O**rganization
DCoD	**D**ata **C**ube **o**n **D**emand
DEM	**D**igital **E**levation **M**odel
EO	**E**arth **O**bservation
EODC	**E**arth **O**bservation **D**ata **C**ube
EPSG	**E**uropean **P**etroleum **S**urvey **G**roup
ESA	**E**uropean **S**pace **A**gency
ESDC	**E**arth **S**ystem **D**ata **C**ube
ESDL	**E**arth **S**ystem **D**ata **L**ab
FORCE	**F**ramework for **O**perational **R**adiometric **C**orrection for **E**nvironmental monitoring
GB	**G**iga**b**yte
GCS	**G**oogle **C**loud **S**torage
GDAL	**G**eospatial **D**ata **A**bstraction **L**ibrary
GEE	**G**oogle **E**arth **E**ngine
GEO	**G**roup on **E**arth **O**bservations

GLANCE	**G**Lobal **LAN**d **C**over and **E**stimation
GPT	**G**raph **P**rocessing **T**ool
HDD	**H**ard **D**isk **D**rive
HPC	**H**igh **P**erformance **C**omputing
IA	**I**ncidence **A**ngle
INPE	**I**nstituto **N**acional de **P**esquisas **E**spaciais (National Institute for Space Research)
ISO	**I**nternational **O**rganization for **S**tandardization
IT	**I**nformation **T**echnology
IW	**I**nterferometric **W**ide Swath
JEODPP	**J**RC **E**arth **O**bservation **D**ata and **P**rocessing **P**latform
JRC	**J**oint **R**esearch **C**entre
JSON	**J**ava**S**cript **O**bject **N**otation
KML	**K**eyhole **M**arkup **L**anguage
LiDAR	**Li**ght **D**etection **a**nd **R**anging
LiMES	**Li**ve **M**onitoring of **E**arth **S**urface framework
LZW	**L**empel-**Z**iv-**W**elch compression
L1AS	FORCE **L**evel-**1** **A**rchiving **S**uite
L2PS	FORCE **L**evel-**2** **P**rocessing **S**ystem
MAJA	**M**ACCS-**A**TCOR **J**oint **A**lgorithm
MSI	**M**ulti**S**pectral **I**nstrument
NASA	**N**ational **A**eronautics and **S**pace **A**dministration
NDVI	**N**ormalized **D**ifference **V**egetation **I**ndex
NRB	**N**ormalized **R**adar **B**ackscatter
ODC	**O**pen **D**ata **C**ube
OGC	**O**pen **G**eospatial **C**onsortium
OLAP	**O**n**l**ine **A**nalytical **P**rocessing
OLI	**O**perational **L**and **I**mager
OSGeo	**O**pen **S**ource **Geo**spatial Foundation
PAA	CARD4L **P**roduct **A**lignment **A**ssessment
PFS	CARD4L **P**roduct **F**amily **S**pecification
QAI	**Q**uality **A**ssurance **I**nformation
SAR	**S**ynthetic **A**perture **R**adar
SDC	**S**wiss **D**ata **C**ube
SDI	**S**patial **D**ata **I**nfrastructure
SNAP	**Sen**tinel **A**pplication **P**latform
SR	**S**urface **R**eflectance
SRTM	**S**huttle **R**adar **T**opography **M**ission
SSH	**S**ecure **S**hell **P**rotocol
STAC	**S**patio**T**emporal **A**sset **C**atalog
TC	**T**opographic **C**orrection

TDC	**T**huringian **D**ata **C**ube
TOA	**T**op **O**f **A**tmosphere
USGS	**U**nited **S**tates **G**eological **S**urvey
UTM	**U**niversal **T**ransverse **M**ercator
VH	**V**ertical transmit, **H**orizontal receive (SAR cross-polarization)
VRT	**V**irtual **R**aster **T**able
VV	**V**ertical transmit, **V**ertical receive (SAR co-polarization)
WCS	**W**eb **C**overage **S**ervice
WCPS	**W**eb **C**overage **P**rocessing **S**ervice
WEF	**W**orld **E**conomic **F**orum
WGS84	**W**orld **G**eodetic **S**ystem 19**84**
WMS	**W**eb **M**ap **S**ervice
WVDB	**W**ater **V**apor **D**ata**b**ase
XML	E**x**tensible **M**arkup **L**anguage
YAML	**Y**AML **A**in't **M**arkup **L**anguage
6S	**S**econd **S**imulation of a **S**atellite **S**ignal in the **S**olar **S**pectrum algorithm

1 Introduction

1.1 Motivation

Earth observation (EO) satellites have been producing diverse and consistent datasets providing valuable information about the Earth's surface for multiple decades now. This continuously growing volume of data and its derived products support the assessment and monitoring of global policy frameworks, such as the United Nations Sustainable Development Goals (Anderson et al., 2017), contribute to the climate data records of several Essential Climate Variables (Hollmann et al., 2013) and assist decision makers in the sustainable use of Earth's resources (Eckman & Stackhouse, 2012).

The accessibility to an extensive volume of EO data has increased continuously, which can be emphasized with two of the most important satellite data archives: the Landsat program by the National Aeronautics and Space Administration (NASA) and the United States Geological Survey (USGS), and the Copernicus program by the European Commission and the European Space Agency (ESA). A free and open access policy for all data produced by the Copernicus program has been agreed upon before the launch of the first Sentinel satellite in 2014 (European Commission, 2013). The decision for such an access policy has been reinforced by reports that forecast significant positive impacts in various socioeconomic areas of the European Union (PricewaterhouseCoopers, 2019). Access to the Landsat archive, including datasets starting as early as 1972, was facilitated by a change to an open data policy in 2009. The policy change subsequently resulted in a significant increase in usage as well as scientific and public value of the data (Wulder et al., 2012; Zhu et al., 2019).

As petabytes of EO and other geospatial data have become available, the term *big Earth data* has been designated and discussions about associated opportunities and challenges have been started (Boulton, 2018; Guo et al., 2016). Challenges related to big data in general have been characterized as *Volume, Velocity and Variety* (Laney, 2001), which has also been adapted in regard to big Earth data and EO data in

particular: "Volume (e.g., data volumes have increased by 10 in the last 5 years); Velocity (e.g., Sentinel-2 is capturing a new image of a given place every 5 days); and Variety (e.g., different type of sensors, spatial/spectral resolutions)" (Giuliani, Camara, et al., 2019, p. 1). This concept of big data challenges has been extended even further by adding the veracity (Saha & Srivastava, 2014) (i.e., concerning data quality) and value of data (Sudmanns et al., 2020). These challenges inhibit the full potential of the data to be exploited by the average user, especially when many still rely on the traditional approach of downloading and processing the data on a local system. Even for small spatial scales, limitations in file storage and computing resources can render the analysis of EO time-series impracticable.

In the past few years, a demand for new and innovative technological solutions has promoted the emergence of various platforms that not only provide access to open and commercial EO data repositories, but also offer processing capabilities via cloud-based infrastructures. Notable examples include Google Earth Engine (GEE) (Gorelick et al., 2017), the JRC Earth Observation Data and Processing Platform (JEODPP) (Soille et al., 2018) and Sentinel Hub (Sinergise Ltd., n.d.). All of these cloud-based management and analysis platforms can represent viable alternatives to the traditional data-centric approach. However, no one-fits-all solution exists and the fact that most platforms either rely on proprietary, closed-source software or necessitate the purchase of storage space and processing resources, can be an important drawback for some user groups.

Another innovative solution that gained popularity among the EO community in recent years is the Open Data Cube (ODC). This open-source software library facilitates the management and analysis of large volumes of EO data and can be deployed on a variety of systems, like High Performance Computing (HPC) systems or cloud environments. The project is supported by various institutions such as Geoscience Australia, the USGS, and the Committee on Earth Observation Satellites (CEOS) (Killough, 2018).

Earth Observation Data Cubes (EODC) based on the ODC software have been successfully deployed for several regions around the world, such as Australia (Lewis et al., 2017), Switzerland (Giuliani, Chatenoux, et al., 2017), Catalonia (Spain) (Maso et al., 2019) and Colombia (Bravo et al., 2017). These national and regional EODCs enable researchers and decision makers to efficiently retrieve information from large EO datasets, while being in control of their own data management and analysis platform. Furthermore, they facilitate the implementation and development of new methodologies for analysis and can be used to present scientific results to the public (e.g., Digital Earth Australia, n.d.).

Giuliani, Chatenoux, et al. (2017) identified data access and data preparation as two major challenges for the implementation of EODCs, both of which regard the generation of Analysis Ready Data (ARD). EO data that has been processed and organized to a minimum set of requirements (e.g., radiometric, atmospheric and geometric corrections) to enable immediate analysis without additional effort, is considered ARD (Lewis et al., 2018). While large volumes of EO data are freely available from data producers, the provision of ARD products that do not necessitate additional processing, is currently lacking behind. As a result, many EODC creators still rely on their own processing approaches (e.g., Giuliani et al., 2018; Ticehurst et al., 2019). New software tools have emerged in recent years that strive to fill this gap and offer a solution to generate ARD independently (e.g., Frantz, 2019; Truckenbrodt, Freemantle, et al., 2019). However, no software tool exists at the moment that can single-handedly gather EO data from different sensor types for a region of interest, apply the necessary processing steps to create ARD products, and finally prepare the products to be used concurrently in a single EODC.

The implementation of EODCs to analyze EO data of large volume and variety, not only has great potential on national but also on smaller spatial scales. The process of policy- and decision-making in federal states can directly benefit from EO derived information. Giuliani, Egger, et al. (2020), for example, have demonstrated how EODCs can support environmental monitoring at local and national scales through the production of essential variables, which are used to observe and monitor the evolution of important Earth system components (e.g., biosphere and hydrosphere).

For the Free State of Thuringia, located in Central Germany, forest ecosystems play an important role and are particularly subjected to climate related impacts (Frischbier et al., 2013). Severe droughts, as experienced by Central Europe in 2003 and 2018, have caused unprecedented tree mortality, not only through heat stress but also secondary effects like forest fires and insect attacks (Schuldt et al., 2020; Senf et al., 2020). A climate bill was enacted in 2018, which includes goals for the adaptation and mitigation of climate related impacts (Freistaat Thüringen, 2018) and could be supported by information derived from an EODC.

The importance of measures that try to alleviate the impacts of a forthcoming climate crisis will continuously grow in the near future. Extreme weather, climate action failure, and human environmental damage were identified by the World Economic Forum (WEF) as major global risks in terms of impact and likelihood (WEF, 2021). The damage caused by extreme weather events, for example, has increased in recent decades across Europe (Kron et al., 2019), and by the year 2100 about two-thirds of the European population could be affected annually by weather-related disasters, such as heatwaves (Forzieri et al., 2017).

The extensive volume of data from EO satellites already helps in better understanding complex natural systems, including the challenges they face and the consequences of human interactions (e.g., de Araujo Barbosa et al., 2015). To continuously derive more valuable information from the data and to make the process itself more efficient, however, further exploration of novel data management and analysis methods is necessary.

1.2 Objectives & Structure

The main objective of this thesis is to implement an EODC for the Free State of Thuringia — the Thuringian Data Cube (TDC). The functionality and usability on a local HPC system is tested with two use cases, one of which relates to possible drought-related impacts in a study site of the Roda forest. A second objective is the creation of a suitable software tool that uses existing open-source software to cover all aspects necessary for the technical implementation. Additional motivations in regard to the development of such a tool are the usability for others and the facilitation of reproducibility. In summary and also in practical order, the objectives of this thesis are:

- Development of a software tool that facilitates all necessary steps for the creation of an EODC on a local HPC system.
- Implementation of the TDC by utilizing the developed tool.
- Assessment of functionality and possible improvements both in regard to the tool and the TDC implementation.
- Demonstration of the usage and potential of the TDC through two use cases.

The structure is as follows: Chapter 2 provides an overview of current literature in regard to the main topics that are covered throughout this thesis, namely Analysis Ready Data and Earth Observation Data Cubes. Chapter 3 describes the development of the software project and all utilized components. The implementation in the form of an EODC for the Free State of Thuringia and demonstrations of its functionality and usage are covered in Chapter 4. The results are discussed in Chapter 5, and finally, Chapter 6 concludes the thesis.

2 State of the Art

2.1 Big Data & Innovation

Since the beginning of the current century, a majority of the world's technological capacity to store, communicate, and compute information has relied on digital formats (Hilbert & Lopez, 2011). This has not only resulted in the amount of data growing continuously, but also the importance of big data being realized in various fields of society (WEF, 2012).

The arising challenges and opportunities of growing amounts of EO and geospatial data have already been important topics in the 1990s. Al Gore coined the term *Digital Earth* in a speech in 1998, where he envisioned a digital representation of our Earth and how it could benefit society (Gore, 1998). In the same decade, the need for improved management strategies of spatial information has increasingly been recognized by administrations worldwide and facilitated the development of Spatial Data Infrastructures (SDIs) (Schade et al., 2019).

As presented by Guo et al. (2016) and Boulton (2018), challenges and opportunities related to the current era of big Earth data are still being discussed two decades later. In the context of EO, this era is particularly being fueled by the open data policies of the Landsat program, which includes an archive of over 40 years of global EO data (Wulder et al., 2019), and the Copernicus program with its family of Sentinel satellites (Aschbacher, 2017).

The volume of available EO and geospatial data keeps growing and the variety of spatial, spectral, and temporal resolutions adds to the challenge of generating valuable information from this data. New missions are soon to be launched (e.g., Kellogg et al., 2020), new sensor technologies are being developed (e.g., Kampe & Good, 2017), and the commercial EO sector is striving to surpass the spatial and temporal resolution of publicly available EO data (e.g., Farquharson et al., 2018).

Innovative trends in the rapidly evolving information technology (IT) sector have frequently been adopted by the geospatial sector. Diaz & Remke (2012) investigated this subject in relation to the development of SDIs and named cloud computing as an important trend. It describes a paradigm that allows individuals to access not only data that is being managed in a remote facility but also applications and computing power over the internet and on demand (Foster et al., 2008).

Sudmanns et al. (2020) describe a change in EO data analysis workflows that is suitable for dealing with large amounts of EO data, where analysis-ready datasets and appropriate tools for information extraction are being provided in a cloud environment. They also indicate that most professionals still rely on the traditional workflow of downloading and processing datasets locally. However, cloud computing platforms such as GEE (Gorelick et al., 2017) have become more popular in recent years and enabled scientists to use EO data in unprecedented global-scale studies (e.g., Hansen et al., 2013).

As a consequence of these changing workflows and increasing volumes of open data being used for analyses, innovation is also happening in closely related areas. Data formats, for example, are evolving and being optimized for network-based access. Yee et al. (2020) compared the data formats GeoTIFF and NetCDF-4, which are two established standards in the domain of Earth Science, with their optimized equivalents Cloud Optimized GeoTIFF (COG) and Zarr. Furthermore, new specifications such as the SpatioTemporal Asset Catalog (STAC) are being developed to provide a standardized way of describing and indexing spatial data and thereby improving its discoverability (Radiant Earth Foundation & Contributors, n.d.-a).

In close relationship to open data and arguably an equally important driver of innovation is open-source software, i.e., collaboratively developed software projects that can be used, changed, and distributed freely (Corbly, 2014). Coetzee et al. (2020) have reviewed the current state of open data and open-source software in the context of the geospatial domain. They came to the conclusion that both "have changed the way in which geospatial data are collected, processed, analyzed, and visualized" (p. 24), and that the combination of open data and open-source software will probably play an even more important role in the future.

2.2 Earth Observation Data Cubes

To address challenges related to big Earth data, new technological solutions have emerged in recent years. Besides cloud computing platforms like GEE, there is an increasing number of software projects in the geospatial domain that in some way or another use the term *Data Cube* and are usually open-source in nature.

There still seems to be a lack of consensus in regard to the exact terminology. However, a number of recent publications are using Earth Observation Data Cube (EODC) as an umbrella term for either such software projects or implementations thereof (e.g., Ferreira, Queiroz, Vinhas, et al., 2020; Giuliani, Masó, et al., 2019; Kopp et al., 2019). According to Killough (2018), they offer a new approach to store, organize, and manage large volumes of analysis-ready EO data and thereby lower the barrier for users to exploit the data to its full potential.

To better understand the fundamentals of EODCs, Section 2.2.1 provides a more comprehensive overview of the concept and how data cubes in EO are defined in literature. Section 2.2.2, on the other hand, gives an overview of important software projects and the Open Data Cube initiative in particular.

2.2.1 Concept

While EODCs have gained popularity in recent years, the underlying idea of data cubes does not originate in the geospatial domain. Already in the early 1990s, data cubes of business and statistical data have been used in the context of Online Analytical Processing (OLAP) and the domain of business intelligence (Nativi et al., 2017). Ariav (1986) defined a data cube as "a three-dimensional and inherently temporal data construct where time, objects, and attributes are the primary dimensions of stored data" (p. 499).

The general notion that data cubes are multidimensional data structures, which include some form of metadata attributes, can be transferred to the concept of data cubes in the geospatial domain. Technological developments related to OLAP data cubes, however, are not directly applicable, as EO data in the form of spatio-temporal rasters is typically densely populated rather than sparsely (Baumann et al., 2019, p. 288).

A simplified visualization of a data cube structure of EO raster data is shown in Figure 2.1 and a matching definition of EODCs as multidimensional arrays of dense raster data is provided by Appel & Pebesma (2019). Fundamentally, they are shaped

by x (longitude; easting) and y (latitude; northing) as the spatial and time as the temporal dimension. Spectral bands or polarization for optical and Synthetic Aperture Radar (SAR) sensors, respectively, are a typical fourth dimension that is not visualized here.

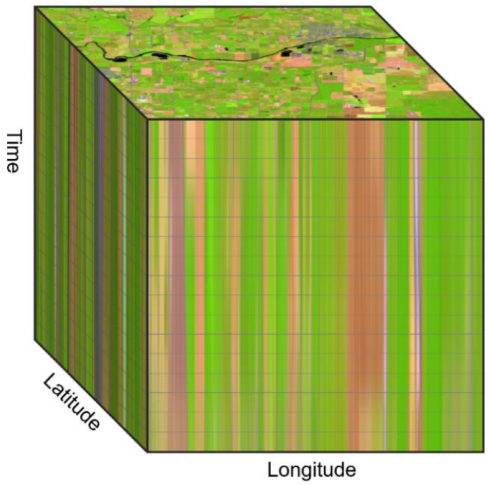

Figure 2.1: Simplified visualization of an EODC as a multidimensional array of dense raster data (Kopp et al., 2019).

In response to inconsistent definitions and terminology, Strobl et al. (2017) proposed a concept that aimed to contribute to a more harmonized definition of what a data cube in the geospatial domain is. They highlight several distinct aspects that need to be considered when creating an EODC and in order to harness its full potential. Matching the cube analogy they identified six aspects, or rather faces, which hereafter are briefly summarized and complemented with relevant publications. The work of Strobl et al. (2017) builds on the conceptual view of the Datacube Manifesto proposed by Baumann (2017). Moreover, Nativi et al. (2017) further expand on the *six faces* concept by introducing a set of modeling views with the goal of emphasizing the interoperability and reusability aspects of data cube infrastructures.

Parameter Model

To understand the information being stored in each dataset of an EODC and to facilitate analysis, the parameter model is describing the semantics of each cell value, including metadata, parameterization, and quality information. Standardization and definition of important elements of the parameter model is done by organizations such as the Open Geospatial Consortium (OGC). It remains a challenge, however, to properly incorporate data of the same kind but from various origins (e.g., surface reflectance data from Sentinel-2 and Landsat 8) because of sensor related differences and the variety of available processing tools and algorithms. This problem can

be mitigated by using ARD as endorsed by CEOS (Lewis et al., 2018), which is presented in more detail in Section 2.3.

Data Representation

Data representation refers to how each axis or dimension of the data cube is encoded. This includes spatial, temporal, spectral, and thematic properties and can be specified by a set of metadata, such as range, scale, and precision. The spatial dimension, for example, is encoded in the form of a grid system, which is based on a geographic projection. The choice of an appropriate projection for the respective region of interest is important, as it can lead to considerable spatial distortion (Steinwand et al., 1995).

Data Organisation

This aspect covers how the data and its cell values are stored. In terms of raster data, this can encompass everything related to the data format (e.g., GeoTIFF, JPEG2000, Zarr), including compression algorithm (e.g., Packbits, Deflate, LZW) and internal partitioning (e.g., band sequential interleaving, block tiling). A comparison of several data formats in the context of cloud storage can be found in Yee et al. (2020), while the influence of compression algorithms in regard to the GeoTIFF format is highlighted by Alberti (2018).

Infrastructure

The infrastructure of storing large volumes of EO data, while ensuring rapid data access and transfer, is another important aspect to consider. EODCs can be implemented on local HPC facilities as demonstrated by Lewis et al. (2017). Cloud computing and storage environments like Amazon Web Services can also be a viable infrastructure option (e.g., Ferreira, Queiroz, Camara, et al., 2020).

Access and Analysis

Appropriate functionalities must be implemented within the infrastructure to access and analyze the stored data and to add new data products to an EODC. The availability of these functionalities to end-users can be provided through APIs (Application Programming Interface), for example. Several software layers (front-end & back-end) that individually cover relevant aspects are also imaginable.

Interoperability

The interoperability between different EODCs is a crucial aspect to prevent them from becoming silos of information. Interoperability can be enabled through the use of widely adopted geospatial standards, which are governed by the OGC and the International Organization for Standardization (ISO). The importance of this aspect is further emphasized by Giuliani, Masó, et al. (2019).

2.2.2 Software Ecosystem

The ecosystem of EODC related projects is diverse and continuously growing. To begin with, it is important to distinguish between EODC related software projects that are usually open-source in nature, and cloud-based processing platforms, whose background is usually commercial. There are recent publications that compare both under an umbrella term (e.g., Gomes et al., 2020). However, a differentiation of both can clear up confusion among users and support a more comprehensible characterization of EODCs. Giuliani, Masó, et al. (2019) discuss several aspects in regard to this distinction and emphasize important limitations of cloud-based platforms that might not be apparent right away, such as the possibility of vendor lock-in.

Also related but not fitting into either category is the openEO API, which pursues the goal of providing a common ground for a variety of back-ends, including those previously mentioned, by connecting them via a multilayered API (Pebesma et al., 2017). The concept behind openEO, which uses a data cube model at its core, is also presented in a recent publication by Schramm et al. (2021).

In the following section, the Open Data Cube is described in more detail, as it plays an important role in the course of this work. An overview of other related EODC projects is provided in Section 2.2.2.2.

2.2.2.1 Open Data Cube

The Open Data Cube (ODC) project originates from the Australian Geoscience Data Cube (AGDC), which initially was developed with the objective to unlock the potential of 27 years of continuous EO data from the Landsat archive covering the entire continent of Australia (Lewis et al., 2016). Major improvements were implemented in the second version of the AGDC (Lewis et al., 2017), and the project was renamed to ODC after long term support was ensured via governance structures (Leith, 2018).

From a technical perspective, the ODC is an open-source software library to access, manage, and analyze large quantities of EO data that can be deployed in a flexible manner (ODC Contributors, n.d.-d). This is achieved through a database, a Python API and a set of command line tools. A more detailed description of the technical background can be found in Section 3.1.3.

Besides being a freely available software library, ODC has developed into a community of people and supporting organizations. Killough (2018) presented the ODC initiative as part of the reorganization from AGDC to ODC, which is supported by

the institutions originally responsible for the AGDC, namely Geoscience Australia and the Commonwealth Scientific and Industrial Research Organization (CSIRO), and additionally CEOS, USGS, and the Analytical Mechanics Associates (AMA). The aim of this initiative is to steward and contribute to the development of the ODC software architecture, thereby enabling its utilization around the world.

In addition to the general ODC initiative, CEOS started a separate CEOS Data Cube initiative based on their goal to "improve data access, data preparation, and data analysis for all global users of satellite data" (Killough, 2018, p. 8630). In this context, the goal was defined to establish operational EODCs based on the ODC software library in 20 countries by 2022.

The Swiss Data Cube (SDC) was one of the first national EODCs and the lessons learned as described by Giuliani, Chatenoux, et al. (2017) have been a valuable resource of information for subsequent deployments (e.g., Asmaryan et al., 2019). The SDC supports the Swiss government in environmental monitoring and reporting, and has been used to monitor the temporal and spatial evolution of snow cover in Switzerland (Dhu et al., 2019, pp. 6–8).

Along with the SDC, the African Regional Data Cube (ARDC) was one of the first EODCs established as part of the CEOS Data Cube initiative. As the name suggests, the ARDC was initially regional in scale and encompassed the countries of Ghana, Kenya, Senegal, Sierra Leone, and Tanzania (Killough, 2019). Since then, the ARDC has developed into a continental-scale data infrastructure project called Digital Earth Africa (Digital Earth Africa, n.d.). According to a report by the WEF, the socio-economic benefits created through Digital Earth Africa could exceed $2bn per year by 2024 (WEF & Digital Earth Africa, 2021).

Another important national EODC deployment that uses ODC at its core is the Brazil Data Cube (BDC) developed by Brazil's National Institute for Space Research (INPE). The methodology of how the BDC was implemented is described in detail by Ferreira, Queiroz, Vinhas, et al. (2020). The BDC has already been used to develop new methods to map land use and cover changes (Santos et al., 2021). Moreover, all software products that are developed as part of the BDC are openly available (Brazil Data Cube Contributors, n.d.) and can be utilized by other projects of the ODC community.

2.2.2.2 Other Projects

EarthServer & BigDataCube

EarthServer, as presented by Baumann et al. (2015), is one of the earliest developed EODC projects. Its approach to serve large volumes of EO data centers on the array database RasDaMan (Baumann et al., 1998) and OGC coverage standards for access and processing, namely Web Map Service (WMS), Web Coverage Service (WCS) and Web Coverage Processing Service (WCPS). The insights gained from EarthServer were refined further with the BigDataCube project, which has been implemented by multiple public and commercial data providers (e.g., CODE-DE and cloudeo AG) to efficiently serve hundreds of terabytes of EO data (Misev et al., 2019).

gdalcubes

The setup of most EODC-related software is not trivial and can limit wider adoption of EODC technology. A project that aims to provide a solution to circumvents this obstacle, is the gdalcubes project (Appel & Pebesma, 2019). It uses on-demand data cubes that are only created when users perform computations on their data. As the name suggests, the widely used Geospatial Data Abstraction Library (GDAL) is a major component, which can handle a large variety of raster data formats (Warmerdam, 2008). The gdalcubes project is available as an open-source C++ library and a package for the programming language R.

Earth System Data Cube & Lab

Mahecha et al. (2020) proposed the concept of Earth System Data Cubes (ESDC) that enable co-interpretation of EO and modelling data (e.g., from climate models). The dimensions of each ESDC, such as spatial, temporal, variable, and frequency, are treated alike and allow the execution of complex workflows by applying user-defined functions. The scientific programming languages Julia and Python are currently supported to work with ESDCs. Furthermore, the Earth System Data Lab (ESDL, n.d.) was introduced by Mahecha et al. (2020), which provides access to curated and analysis-ready ESDCs stored in a cloud environment.

xcube

Similar to ODC, the open-source project xcube is using the Python packages Xarray and Dask as its core packages (xcube Contributors, n.d.). Beyond that, the conceptual direction of xcube is different, as it heavily relies on the data format Zarr to enable the creation of self-contained data cubes. These can then be published and used in a cloud environment, which has been implemented by the commercial Euro Data Cube service (Euro Data Cube Consortium, n.d.).

TileDB

TileDB is an open-source software system managed by Intel Labs and the Massachusetts Institute of Technology. It poses as a universal data engine and data management solution by providing not only a storage engine, but also a data model and data format (TileDB, Inc., n.d.). TileDB can handle sparse and dense arrays of arbitrary dimensionality and therefore facilitates usage in a wide variety of large-scale scientific applications (Papadopoulos et al., 2016). An example on how TileDB can be used to manage large volumes of SAR data is presented by Barker (2020).

2.3 Analysis Ready Data

The aspect of having EO data that was consistently preprocessed and is suitable for time-series analysis is fundamental for the creation of EODCs. This aspect was described by Strobl et al. (2017) as the first *face* of the data cube, and the importance is further emphasized by how closely tied the term Analysis Ready Data (ARD) is being used in the context of EODCs (e.g., Baumann et al., 2019; Giuliani, Chatenoux, et al., 2017; Killough, 2019). When working with EO data, users typically have to deal with the difficulties of data access, data preparation and efficient analysis, all of which can be eased by the generation of ARD (Giuliani, Chatenoux, et al., 2017, p. 102).

Currently, there still is an ongoing discussion about the definition of ARD and agreed specifications for the various satellite sensors available. The issue of ARD being interpreted differently by institutions and data providers is highlighted by Sudmanns et al. (2020) in the context of data access. They come to the conclusion that the potential of ARD can be unlocked by "either finding a common understanding or communicating clearly the ARD's differences, individual characteristics, and suitability for tasks" (p. 845). The CEOS Analysis Ready Data for Land initiative (CARD4L) presents an important step towards a consensus among EO data providers and users, which is further elaborated on in Section 2.3.1.

Access to ARD products is improving continuously. The Landsat archive, for example, is being reprocessed into a tiered Collection 2 product inventory that is consistently calibrated over time (Dwyer et al., 2018; Masek et al., 2020), and effort is being made to create a harmonized Landsat 8 and Sentinel-2 product (Claverie et al., 2018). However, the creation of EODCs is still often linked to alternative approaches of generating ARD to meet specific project requirements (e.g., Ferreira,

Queiroz, Vinhas, et al., 2020). Some of these approaches are briefly described in Section 2.3.2.

In the future, it will be necessary that the burden of generating ARD in a form that is suitable for as many users as possible lies on data providers who have the appropriate facilities to process large volumes of EO data (Giuliani, Masó, et al., 2019, p. 19). Beyond the simple provision of ARD products, Abernathey et al. (2020) argue that current infrastructure challenges of data-intensive scientific fields could be overcome by the combination of analysis-ready, cloud-optimized (ARCO) data repositories and scalable data-proximate computing (i.e., cloud computing).

2.3.1 CARD4L

2.3.1.1 Initiative

Through the CARD4L initiative, CEOS is working to "enable non-expert users access to products that have been processed 'far enough' to be suitable for immediate analysis for a range of applications, while ensuring they are not too specific to only be used for particular topics or areas." (Lewis et al., 2018, p. 7407). CARD4L aims to enable an increase of EODC usage and interoperability, and has set up a framework to achieve the initiative's goals consisting of three core components: Definition, Product Family Specification (PFS), and Product Alignment Assessment (PAA).

The non-prescriptive definition of CARD4L is as follows: "CEOS Analysis Ready Data for Land (CARD4L) are satellite data that have been processed to a minimum set of requirements and organized into a form that allows immediate analysis with a minimum of additional user effort and interoperability both through time and with other datasets." (Lewis et al., 2018, p. 7408).

An overview of the other two components, PFS and PAA, is also provided by Lewis et al. (2018). The PFSs identify comparable and fundamental measurement types that are sensor-agnostic (e.g., surface reflectance), instead of the traditional approach of focusing on unique data streams produced by individual sensors. Moreover, requirements for the generation of ARD are defined by the PFSs. The PAAs on the other hand, are supposed to help data producers in self-assessing the alignment of their products with the PFSs. In addition, PAAs facilitate the provision of independent assessments and peer-reviews.

2.3.1.2 Product Family Specifications

The details provided by the PFSs can be used by data producers to identify which measures are necessary in order to deliver CARD4L compliant ARD to the users. These are further categorized as *Threshold* and *Target* requirements that provide specific information on the categories: general metadata, per-pixel metadata, radiometric and atmospheric corrections, and geometric corrections (Siqueira et al., 2019).

Four PFSs are presently devised: Surface Reflectance (v5.0) and Surface Temperature (v5.0) for optical sensors, and Normalized Radar Backscatter (v5.0) and Polarimetric Radar (v3.0) for SAR sensors (CEOS, n.d.-b). A brief introduction of the Surface Reflectance and Normalized Radar Backscatter PFSs (SR-PFS and NRB-PFS, respectively) follows, as relevant ARD products are produced and used in this work.

General Metadata

At a basic level, both specifications require the provision of general metadata that allows users to assess if the dataset is suitable for their analysis. This includes information about the source data, such as temporal and spatial coverage, processing details, and other important attributes (e.g., spectral bands for SR-PFS, polarizations and orbit direction for NRB-PFS).

Per-pixel Metadata

Both specifications also require per-pixel metadata that can be used to discriminate between individual observations. This includes commonly used masks for the SR-PFS (e.g., cloud and cloud shadow coverage). The NRB-PFS, on the other hand, specifies the requirement for a supplementary local incident angle image, which is based on the digital elevation model used for processing.

Corrections

Geometric corrections are required to be performed for both specifications in order for measurements to be accurately geolocated and comparable through time. Other corrections, however, are sensor specific: radiometric and atmospheric corrections in case of the SR-PFS and radiometric terrain correction for the NRB-PFS, which result in surface reflectance and normalized backscatter intensity measurements, respectively.

2.3.2 Processing Methods

The national and regional EODCs established in relation to the CEOS Data Cube initiative mentioned in Section 2.2.2.1 have developed different approaches for generating their ARD products. To provide both the Swiss and the Armenian Data Cube with ARD, for example, the Live Monitoring of Earth Surface (LiMES) framework was utilized (Asmaryan et al., 2019; Giuliani, Chatenoux, et al., 2017), which is presented by Giuliani, Dao, et al. (2017). Other approaches are described in detail, i.a., by Lewis et al. (2017) for the AGDCv2 and by Ferreira, Queiroz, Vinhas, et al. (2020) for the BDC.

Even though a variety of approaches are used, the common ground usually lies in open-source software and algorithms. Especially for optical sensors an extensive selection of peer-reviewed algorithms for radiometric and atmospheric corrections, as well as commercial and open-source software solutions, exist. Sen2Cor (Main-Knorn et al., 2017), for example, is commonly used to process Sentinel-2 data to a Level-2A Bottom Of Atmosphere (BOA) product, while Lonjou et al. (2016) developed the MACCS-ATCOR joint algorithm (MAJA), which can be used to process both Sentinel-2 and Landsat 8 scenes. The 6S algorithm (Vermote et al., 1997) is implemented in the open-source software ARCSI (Bunting & Contributors, n.d.), which is able to handle a variety of optical sensors. And a software framework that not only focuses on the application of correction algorithms, but also on other important aspects when generating ARD for EODCs (e.g., appropriate tiling and gridding scheme), is the Framework for Operational Radiometric Correction for Environmental monitoring (FORCE) (Frantz, 2019).

In the context of SAR sensors on the other hand, the selection of algorithms to produce an NRB product is more uniform, and ultimately it is rather a choice of which software to use. Some alternatives include the open-source Orfeo Toolbox (Inglada & Christophe, 2009) and the proprietary GAMMA software (GAMMA Remote Sensing, n.d.). ESA's Sentinel Application Platform (ESA, n.d.-c) and more specifically its Sentinel-1 toolbox, is a widely adopted and freely available software to process data from various ESA and third-party SAR sensors. Truckenbrodt, Freemantle, et al. (2019) and Ticehurst et al. (2019) have investigated the feasibility of generating Sentinel-1 ARD products using SNAP and GAMMA and came to a confirming conclusion in both cases.

3 Technical Development

3.1 System Components

In the following Section, individual software components and specific functionalities are introduced that were integrated into the software tool ARDCube, which was developed in the course of this work. The resulting tool was utilized for the implementation of the Thuringian Data Cube and is presented in Section 3.2.

3.1.1 pyroSAR

pyroSAR is an open-source Python package (Truckenbrodt & Contributors, n.d.-a) with the aim to provide a unified and scalable framework for the organization and processing of Sythetic Aperture Radar (SAR) satellite data. As described by Truckenbrodt, Cremer, et al. (2019), it consists of three main components: (1) identification and reading of SAR scenes from various missions, as well as the extraction and homogenization of metadata; (2) organization of the available information in a data archive, either as a SpatiaLite or a PostgreSQL database; (3) utilization of the SAR processing software SNAP and GAMMA, and handling of any required ancillary data. Out of these, the functionality provided by the processing component was utilized for the development of the software tool. More specifically, to handle two particular tasks:

- Creating a Digital Elevation Model (DEM) mosaic.
- Processing of SAR scenes to ARD products using the integrated SNAP API.

Two auxiliary tools are offered by pyroSAR to create a DEM for an area of interest (Truckenbrodt & Contributors, n.d.-b). First, all relevant tiles are obtained from one of various sources, such as the Shuttle Radar Topography Mission (SRTM) in 1 arcsecond (~30 m) or 3 arcsecond (~90 m) spatial resolution. The individual tiles are then merged and clipped to the extent of the area of interest. In addition to the

processing of SAR scenes, the resulting DEM file can also be used by other software components and workflows.

For the processing of SAR scenes, pyroSAR's SNAP API was chosen because of the open-source availability of the SNAP software in comparison to the proprietary alternative GAMMA. Processing via the SNAP API is achieved by adapting XML (Extensible Markup Language) workflows based on user input and then parsing them directly to SNAP's Graph Processing Tool (GPT) (Truckenbrodt, Cremer, et al., 2019). Moreover, the custom workflows can be stored alongside the output files to facilitate the reproducibility of processing results.

The feasibility to produce SAR backscatter products of an appropriate quality for time-series analysis with the SNAP software was investigated by Truckenbrodt, Freemantle, et al. (2019). The study particularly focused on producing Sentinel-1 ARD in regard to EODCs, and ultimately came to an affirming conclusion, which further supports the implementation of pyroSAR's SNAP API into the tool.

3.1.2 FORCE

The Framework for Operational Radiometric Correction for Environmental monitoring, or FORCE in short, is an open-source software project developed by Frantz & Contributors (n.d.-d). The aim of FORCE is to provide a comprehensive solution for processing data from the Landsat and Sentinel-2 archives to ARD, and furthermore, higher-level products.

The software components of FORCE are organized in the form of a level system: level-1 to automatically acquire scenes from data providers, level-2 for processing data to an ARD format, and level-3 for any higher-level processing (e.g., computation of best-available-pixel composites). The Level-1 Archiving Suite (L1AS) and the Level-2 Processing System (L2PS) are two of the main components of FORCE. They constitute an interconnected workflow, which is shown in Figure 3.1 and represents the functionality integrated into the developed software tool.

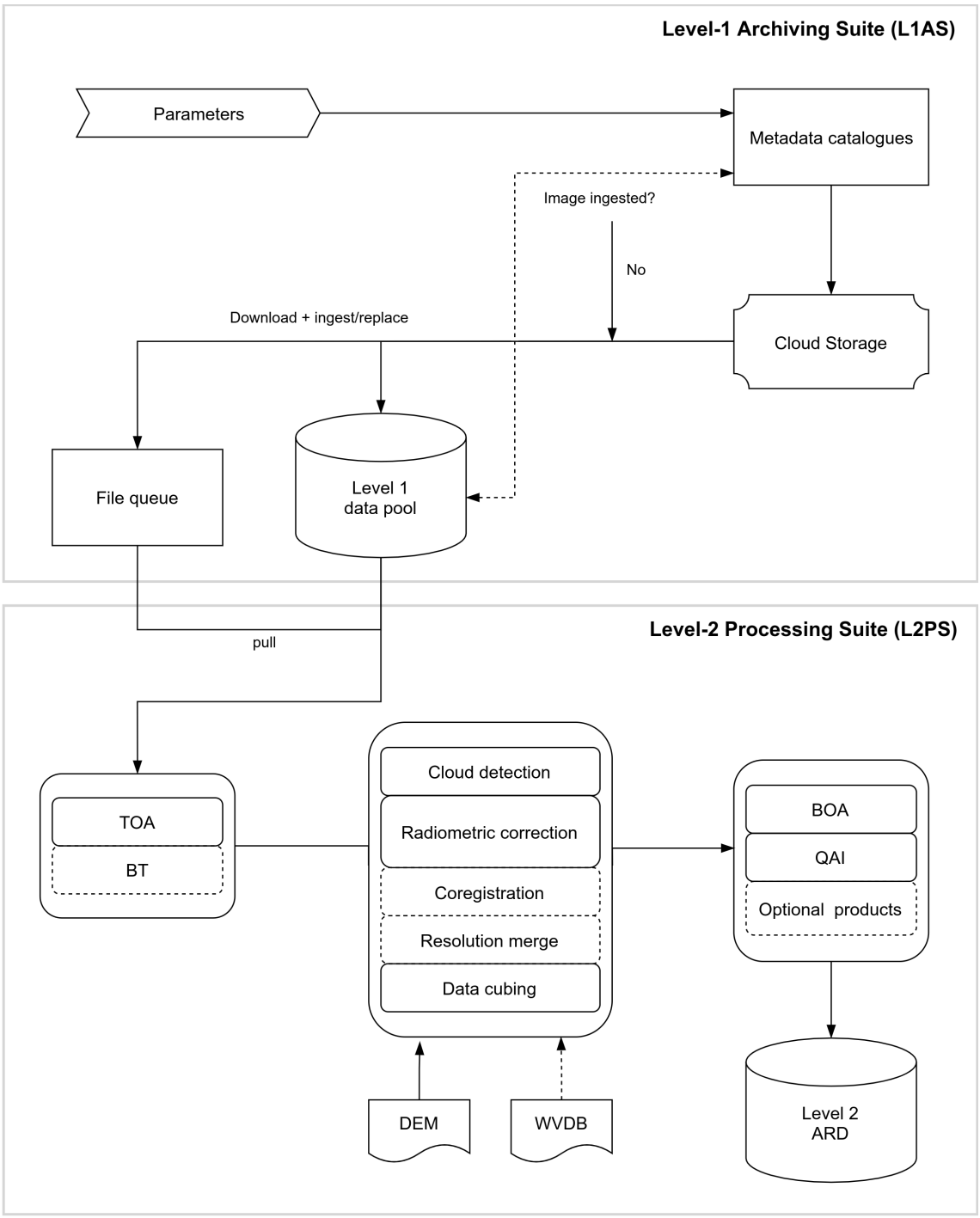

Figure 3.1: FORCE Level-1 Archiving Suite (L1AS) and Level-2 Processing System (L2PS) workflows. TOA = Top Of Atmosphere; BT = Brightness Temperature; BOA = Bottom Of Atmosphere; QAI = Quality Assurance Information; DEM = Digital Elevation Model; WVDB = Water Vapor Database. Adapted from Frantz (2019) and Frantz & Contributors (n.d.-b).

At first, the L1AS component is using provided parameters to query a cloud storage provider (e.g., Google Cloud Storage) and download any requested Sentinel-2 or Landsat archive data. This workflow also checks for already existing scenes so only missing or new scenes are downloaded. A simple text file is used to track the scenes and enqueue them for subsequent processing (*File queue* in Figure 3.1).

The L2PS component is using a processing workflow based on the framework presented by Frantz, Roder, et al. (2016). It includes a modified version of the Fmask code (Zhu & Woodcock, 2012) for cloud masking and the generation of bit-wise Quality Assurance Information (QAI), as well as a radiometric correction with radiative-transfer-based atmospheric correction (Tanré et al., 1990, 1979). Furthermore, the workflow includes the option to utilize one of three implemented algorithms to improve the spatial resolution of the 20 m Sentinel-2 bands. Finally, the data is brought into a data cube appropriate format, i.e., reprojected and split into non-overlapping image chips based on a custom grid of rectangular tiles. A more comprehensive description of the processing workflow, including options not implemented here, can be found in Frantz (2019) and Frantz & Contributors (n.d.-b).

In addition to the L1AS and L2PS components, some auxiliary modules of FORCE are utilized as well, namely *force-tabulate-grid* and *force-mosaic*. The former creates a KML (Keyhole Markup Language) file of the tiling grid, and the latter automatically creates mosaics in the VRT (Virtual Raster Table) format for all image chips of the same date. Both outputs are for visualization purposes only. A third auxiliary module, *force-cube*, serves a more important task, as it is used to bring processed SAR data into the same data cube appropriate format mentioned above. Thereby facilitating the concurrent use of optical and SAR datasets in the same EODC.

3.1.3 Open Data Cube

As described in Section 2.2.2.1, at its core the Open Data Cube (ODC) is an open-source software library aimed at the management and analysis of large volumes of EO data, which has successfully been leveraged to create EODCs of national (Giuliani, Chatenoux, et al., 2017) and even continental scales (Lewis et al., 2017). The main technical components of ODC are a database for data management, and a Python based API to query and access the data.

ODC currently uses a PostgreSQL database to catalog information about all EO data that is available for a given deployment. This allows large datasets to be queried (e.g., by time and location) initially without having to access the actual storage location (Leith, 2018). As described by ODC Contributors (n.d.-d), the process of cataloging information in the context of ODC is also called *Indexing* and is achieved in two separate steps as shown in Figure 3.2.

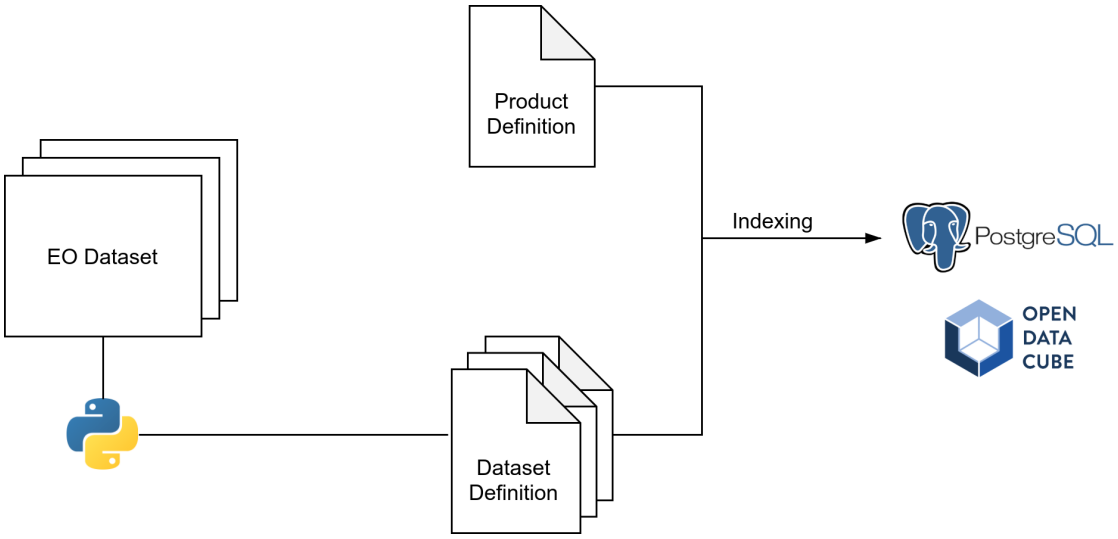

Figure 3.2: ODC indexing workflow. A Python script is commonly used to generate dataset definition documents for each scene in a given EO dataset, whereas a product definition document needs to be created manually for the same dataset. All documents are stored in the YAML format. The indexing step itself can be done via the command-line. Adapted from ODC Contributors (n.d.-d).

First, a product definition document is manually created for a given dataset. It is stored in the YAML (YAML Ain't Markup Language) format and contains general information that is common for each scene of a dataset. This can include general metadata, storage information like Coordinate Reference System (CRS) and spatial resolution, as well as measurement descriptions. The measurement descriptions can also be extended from simply listing which spectral or polarization bands each scene includes, to providing bit-level descriptions of quality flags, such as the QAI bands generated by FORCE. This information can then be used during analysis to easily create masks and exclude undesirable pixel values (e.g., cloud covered pixels). An example product definition document is provided by (ODC Contributors, n.d.-e).

During the second step, dataset definition documents are created for each individual scene in a given dataset. This step is also called *Data Preparation* and can be automated using customized Python scripts. For each scene, metadata that is specific for the given scene is extracted and also stored in a YAML file. Most importantly this includes storage location on the local file system, as well as acquisition time and georegistration information to enable temporal and spatial querying, respectively. Besides additional metadata, provenance can also be stored and tracked, provided that the source dataset was indexed as well. An example dataset definition document is provided by (ODC Contributors, n.d.-c).

In addition to indexing, it is also possible to *Ingest* datasets. This process converts indexed datasets to a new file format and structure, such as NetCDF. This can improve the efficiency of data access during analysis. However, this optional step

was not explored further in the course of this work. Based on responses in the ODC community, it is neither commonly being used, nor actively being developed anymore (Woodcock & Kouzoubov, 2020).

After the database has been made aware of available datasets via the process of indexing product and dataset definition documents, it can be used to query and retrieve data for analysis. An open-source Python package is available for this purpose (ODC Contributors, n.d.-a), which leverages the packages Xarray (Hoyer & Hamman, 2017) and Dask (Rocklin, 2015) as two of its main dependencies. While Dask is providing a framework for parallel computing, Xarray is a toolkit for working with data that is structured as multidimensional arrays and closely integrated with the former. The combination of Xarray and Dask is also being utilized in other projects that deal with large volumes of geoscientific data, such as Pangeo (Pangeo Contributors, n.d.-b).

When data is loaded into a Python environment it is organized as an Xarray Dataset including separate data variables for each requested band and appropriately labelled temporal and spatial dimensions (ODC Contributors, n.d.-d). The evaluation of arguments (e.g., loading of data or executing a computation) is also supported in a lazy manner, which means that each evaluation only happens when the value of an argument is first demanded (Bloss et al., 1988, p. 147). In combination with the chunking of data into smaller portions, it is possible to run algorithms over very large EO datasets efficiently, even if the data itself does not fit into the system's memory.

Some additional utilities are provided with the core Python package of ODC, like a masking tool to create Boolean masks from bitwise QAI products during analysis. More extensive collections of tools and algorithm examples are available in repositories created by users of the ODC community (e.g., Krause et al., 2021). Furthermore, a selection of applications has been developed that extend the functionality of ODC in various ways (ODC Contributors, n.d.-b).

3.1.4 Containerization

Encapsulating a software environment, including its dependencies, system libraries, settings, and runtime code, so that it can be run on different computing environments reliably, is known as software containerization (Docker, n.d.). At the same time as many scientific domains increasingly rely on computational work using widely adopted programming languages like Python and R, the challenge of reproducibility is becoming more important and can be addressed by utilizing emerging

containerization technologies, such as Docker (Boettiger, 2015). The implementation of this aspect played an important role in the development of the software tool for various reasons:

- Simplifying the process of installing and setting up the required software in general and regarding the deployment on HPC systems in particular.
- Avoiding possible issues related to the concurrent use of software components that rely on similar dependencies.
- Facilitating reproducibility by enabling data processing workflows that can be replicated much easier and more reliable in contrast to the alternative approach of not using containerized software components.

The containerization solution implemented here is Singularity, which is open-source and was created with the motivation to provide a solution that is suitable for scientific applications on HPC systems (Kurtzer et al., 2017). Even though Docker is a containerization platform that is widely adopted in the IT sector, the requirement for high-level access privileges when running containers poses a security concern for HPC administrators (Kurtzer et al., 2017; Silver, 2017). The architecture of Singularity containers as described by Kurtzer et al. (2017), on the other hand, constitutes no such concern.

A Singularity container can be created from a definition file that provides a kind of recipe for what should be installed and included inside the container. Alternatively, they can be created from existing Docker container images and the official Docker registry, which in combination with the fact that the resulting containers exist as individual and moveable files, offers high flexibility. As shown by Garofoli et al. (2019), Singularity can be used to create containerized scientific applications that are portable, scalable, and ensure reproducible results.

3.2 System Architecture

The open-source software project called *ARDCube* was developed in the course of this work, which integrates all components introduced in Section 3.1. It is hosted on GitHub (https://github.com/maawoo/ARDCube) and can be used by others without restrictions as stated by the MIT license. An overview of all integrated components, as well as the main workflows managed by individual modules, is shown in Figure 3.3. The following sections describe the most important aspects in more detail. Documents that are more specific, such as setup instructions, will be provided in the GitHub repository.

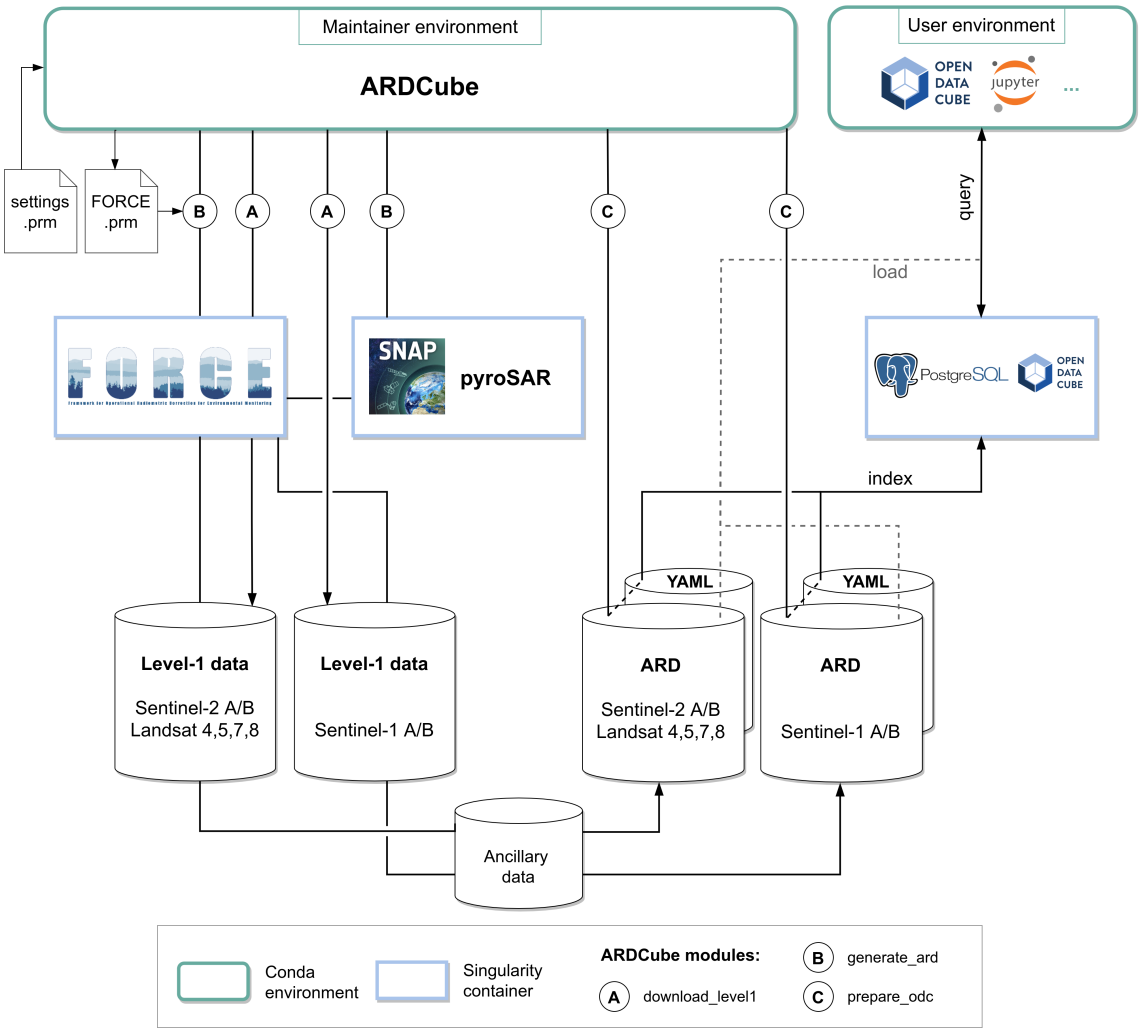

Figure 3.3: Overview of the ARDCube tool, its modules and the individual workflows it manages. The modules are leveraging functionalities provided by the containerized software components to download optical and SAR level-1 data (A), process the data to ARD products (B), and prepare metadata documents in the YAML format for each scene of a given ARD product (C). The YAML documents are then indexed into a PostgreSQL database populated with an ODC schema in order for the ODC Python API to access the ARD products.

3.2.1 Python Framework

A variety of Python packages form the basic framework of ARDCube. They are utilized to manage the integrated software components and further extend their functionality by enabling the automation of specific workflows. To manage these Python packages, on the other hand, the package manager Conda (Anaconda, Inc., 2017) is used. It offers the possibility to create self-contained environments that can include a custom selection of Python packages. In addition, these environments can easily be shared and replicated by others, which yet again facilitates reproducibility. Based on the prospect of how ARDCube could be used in the future, two user groups were identified and hence all packages are managed in two separate Conda environments:

- A maintainer environment, which includes all packages necessary for the automated workflows to download, process, and organize EO datasets.
- A user environment, which includes packages necessary for analyzing the data.

Some important packages used in the maintainer environment are introduced at relevant points in the course of the following sections. The user environment, on the other hand, is intended to be used more flexibly. At a very basic level, only the ODC Python package and its dependencies (see Section 3.1.3) would be needed to access datasets that were indexed into an ODC database. However, other packages are recommended to be included. JupyterLab, for example, allows users to start interactive computing environments in a remote web browser, which can be very useful in the context of working on HPC systems (Thomas et al., 2021). In the end, extending the environment is left to each user and can be as simple as packages for the visualization of results, or more advanced with packages such as Numba (Lam et al., 2015) that can be used to optimize the performance of array-based computations.

3.2.2 Usage & Parameterization

The usage of ARDCube and its underlying, containerized software components is primarily limited by available computing resources, especially in regard to processing large volumes of EO data to ARD products. A second limitation is the installation of Singularity, as ARDCube is built around the containerization it provides. For an installation of Singularity on MacOS or Windows systems, additional steps via a lightweight virtual machine are needed (Sylabs, Inc., n.d.), which has not been

tested in the course of this work. HPC systems, however, are usually running a Linux distribution, which Singularity runs natively on, and provide a reasonable amount of computing resources. Therefore, ARDCube is intended but not limited to being used as a command-line application on HPC systems.

The individual software components use different ways to parameterize their processes. Therefore, a solution was needed that streamlines the procedure for an ease of usage in general, while leaving the option to access the native parameterization if a more advanced usage is required. The approach that was ultimately implemented relies on the configparser module (configparser Contributors, n.d.), which is part of Python's standard library. A selection of important parameters is listed in a single parameter file, *settings.prm*, that need to be filled by the user prior to using the ARDCube modules. The parameters are listed as key-value pairs and grouped into relevant categories (e.g., download and processing parameters). The entire range of available parameters, however, is not hidden but easily accessible and adjustable, as relevant files (e.g., the original FORCE parameter file) are located in subdirectories relative to where the ARDCube parameter file is stored.

3.2.3 Modules

The main modules of ARDCube and the underlying workflows are briefly introduced in the following. As already mentioned, Singularity containers that each include an individual software component, provide the core functionality for most of the workflows. To access these from Python scripts and execute specific processes inside the containers, the package spython (Sochat & Contributors, n.d.) is utilized throughout the modules.

Supplementary to the main modules, various utility functions are provided by ARDCube. These include, for example, the automatic creation of a DEM mosaic for an area of interest as mentioned in Section 3.1.1, and the handling of auxiliary functionalities related to FORCE and ODC.

download_level1

The module *download_level1* is used to acquire level-1 EO data from data providers. The retrieval of optical satellite data, i.e., the Landsat archive and Sentinel-2, is achieved through the *force-level1-csd* module of FORCE, which is part of the L1AS (see Figure 3.1) and currently accesses Google Cloud Storage (GCS) to download data. To query for available data, the module relies on local copies of the metadata catalogs for each GCS dataset. As shown in Figure 3.1, only missing or incomplete scenes are downloaded, which means that the process can be cancelled and then

resumed at a later time. Furthermore, the processing baseline of Sentinel-2 scenes is checked so that only versions with the highest baseline or latest processing date are acquired (Frantz & Contributors, n.d.-a), therefore ensuring the highest available data quality.

Download of SAR data is covered by the Python package sentinelsat (Wille et al., n.d.). However, it can only be used to acquire Sentinel-1 scenes stored on the Copernicus Open Access Hub. Downloading data from other SAR satellite missions is currently not implemented and has to be done manually from relevant sources. Similar to *force-level1-csd*, incomplete downloads are continued and complete files are skipped.

generate_ard

The module *generate_ard* controls all workflows related to processing level-1 data to ARD products. For optical satellite data the L2PS workflow described in Section 3.1.2 is utilized with the FORCE container. An important aspect for this workflow is the proper handling of parameterization, as FORCE offers an extensive selection of options that are controlled via a parameter file. Parameterization has already been briefly described. In this particular case, however, it is important to note that the native parameter file of FORCE is used as a template and filled with relevant user provided parameters from the ARDCube parameter file mentioned in Section 3.2.2. The template is not overwritten by the ARDCube scripts but rather saved as a timestamped copy every time the processing workflow is executed. Thereby, any changes to the default processing parameters can afterwards be retraced more easily.

The processing of SAR satellite data is achieved via pyroSAR's SNAP API, as described in Section 3.1.1. This produces radiometrically terrain-corrected gamma nought backscatter data, which is then further altered with two post-processing steps. First of all, the Python package rasterio (MapBox, Inc. & Contributors, n.d.) is used to crop the output files to the extent of the area of interest, including outer boundaries of no data values to mitigate data redundancy. Secondly and as mentioned in Section 3.1.2, *force-cube* is used to reproject the files and create non-overlapping tiles based on a predefined grid. This step is intended to bring the SAR data into the same format as optical data that has already been processed. In this case a FORCE-internal datacube definition file containing the relevant gridding information has been automatically created and can easily be used with *force-cube*. Alternatively, a template for this definition file is provided, which can be adjusted manually if, for example, a SAR-only EODC is intended to be created with ARDCube.

prepare_odc

Finally, the module *prepare_odc* can be used to automatically create the dataset definition documents for each file of a given dataset. These are necessary in order for datasets to be indexed into an available ODC database, as mentioned in Section 3.1.3. Hereby, mostly Python packages of the standard library are used to collect the necessary information and create the YAML files. Additionally, the aforementioned rasterio package is used to collect spatial information about each raster file, which is an essential aspect of each definition document. The generated documents are stored with the same naming scheme and in the same location as each associated source raster file.

4 Thuringian Data Cube

4.1 Implementation

With the help of the developed software tool ARDCube, the Thuringian Data Cube (TDC) was implemented on the HPC system TerraSense, which is used by the Department of Earth Observation at the Friedrich Schiller University Jena. It runs CentOS 7 as the operating system and the available computing resources on the selected node consist of four AMD Opteron 6348 processors that operate at 2.8 GHz by default and contain 12 single-threaded cores each. Moreover, 500 GB of computing memory and an HDD (Hard Disk Drive) file system are available. The following sections describe the implementation in reference to the ARDCube modules used.

4.1.1 Data Selection

For the initial implementation of the TDC, EO datasets were acquired for a selected area and timeframe. The spatial extent of the area of interest covers the Free State of Thuringia, located in central Germany (Figure 4.1), while a timeframe of three years was chosen for the temporal extent: from 2017-01-01 until 2019-12-31.

Based on the spatial and temporal extents, EO data for the optical satellites Landsat 8 and Sentinel-2A/B, were acquired using the *download_level1* module of ARDCube. Both Landsat 8 and Sentinel-2A/B carry multispectral sensors: OLI (Operational Land Imager) and MSI (MultiSpectral Instrument) for Landsat 8 and Sentinel-2A/B respectively. They work passively by measuring particular parts of the electromagnetic spectrum that is emitted by the sun (i.a., infrared and visible light) and reflected back from the Earth.

Furthermore, EO data for the Sentinel-1A/B satellites was already available on TerraSense for the same extents. In contrast to the optical satellites, Sentinel-1A/B use a C-band SAR instrument to actively send and receive signals to collect information about the Earth's surface. The data was acquired in the Interferometric

Wide Swath (IW) acquisition mode, for both ascending and descending orbits, and include both VH (Vertical transmit; Horizontal receive) and VV (Vertical transmit; Vertical receive) polarizations.

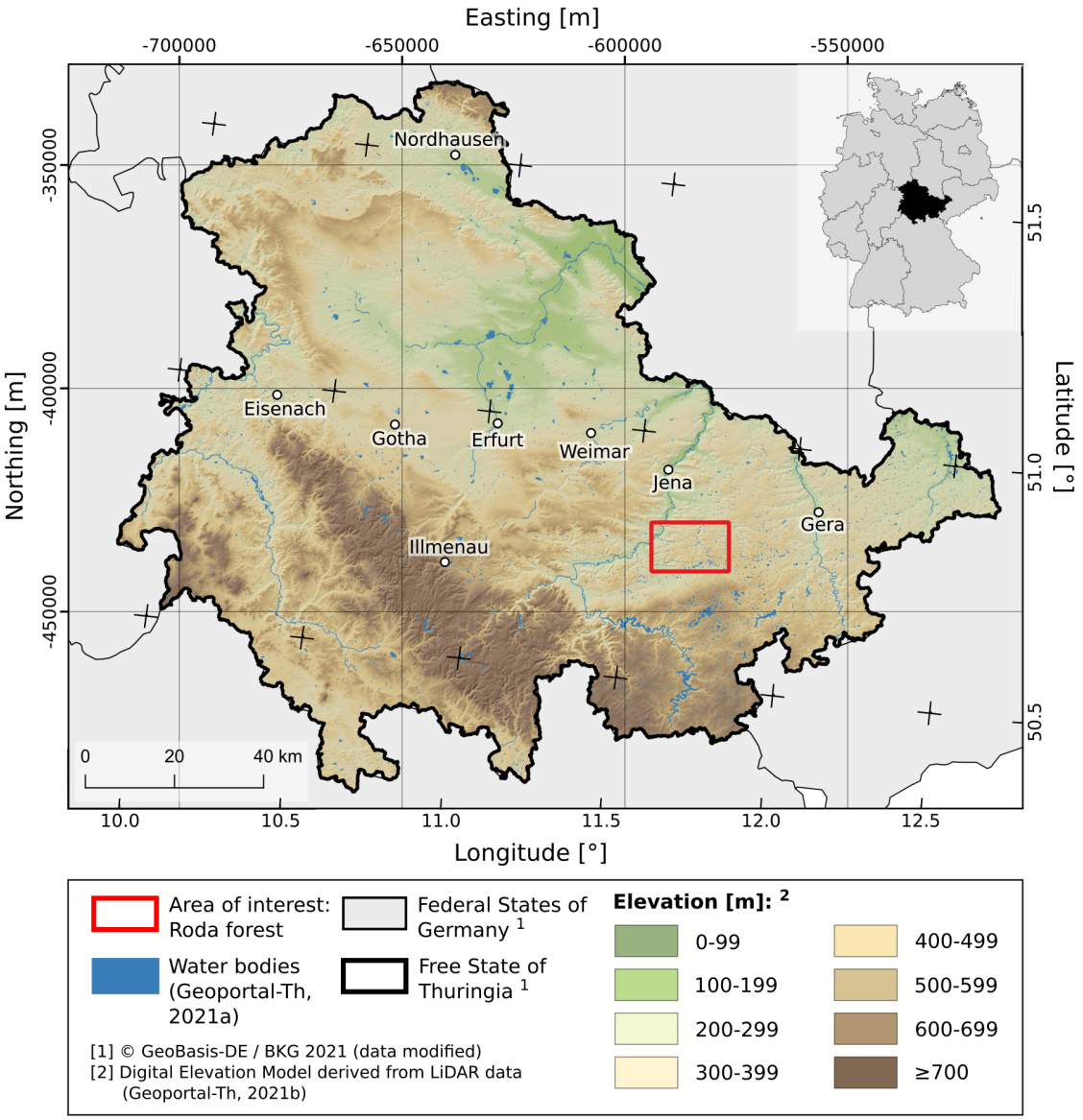

Figure 4.1: Extent of the Free State of Thuringia, including various characteristics (e.g., location in Germany, topography and the location of major cities). An area of the Roda forest is highlighted, which is of interest for the use case described in Section 4.2.2. The map is projected in the GLANCE7 EU grid (Holden, n.d.) with corresponding easting and northing coordinates. An additional grid shows latitude and longitude coordinates of the commonly used WGS84 reference system (EPSG:4326).

Additional maps are available in Appendix A with the tiling schemes for Sentinel-2A/B (Figure A-1) and Landsat 8 (Figure A-2) acquisitions overlaid over the area of interest. For Sentinel-1A/B, on the other hand, no fixed tiling scheme is used. ESA regularly provides acquisition segments covering a period of 12 days

each, which is not feasible to visualize in this case. However, exemplary scene footprints for ascending and descending orbits are provided in Figure A-3.

Hereinafter, references of Sentinel-2 and Sentinel-1, always include each mission's currently active satellites (i.e., Sentinel-2A and -2B; Sentinel-1A and -1B). Any reference to an optical dataset includes both Landsat 8 and Sentinel-2 data.

4.1.2 Optical Satellite Data

The level-1 data acquired were processed to an ARD format using the *process_ard* module of ARDCube. In total, 234 Landsat 8 scenes with a size of 247 GB, and 2200 Sentinel-2 scenes with a size of 1400 GB formed the basis for this particular step of the TDC implementation.

The processing workflow of the FORCE L2PS module (Figure 3.1) can be customized with various processing parameters. In the case of processing data for the TDC, mostly default settings were chosen, as they are commonly used by the FORCE developers themselves to generate ARD (Frantz & Contributors, n.d.-c). Some of the more important parameters are listed in Table 4.1.

To perform the topographic correction, as well as improving cloud and cloud shadow detection and atmospheric correction, a Digital Elevation Model (DEM) is needed. A DEM with 10 m spatial resolution for the entire extent of the Free State of Thuringia was provided by the Department of Earth Observation for this purpose, which has been derived from openly available LiDAR (Light Detection and Ranging) data (Geoportal-Th, 2021b). The DEM was used in an uncompressed format to prevent a possible negative impact on processing performance (cf. Alberti, 2018).

No water vapor correction using an external water vapor database was performed, as this option is only relevant for Landsat data and in particular Landsat 4-7 (Frantz et al., 2019), which were not used for this implementation. The Improphe algorithm was used to improve the spatial resolution of the 20 m Sentinel-2 bands during the Resolution Merge processing step. Further details on the algorithm are provided by Frantz, Stellmes, et al. (2016).

Table 4.1: FORCE L2PS parameters used to process Landsat 8 and Sentinel-2 scenes for the implementation of the TDC. BRDF = Bidirectional Reflectance Distribution Function; AOD = Atmospheric Optical Depth.

Parameter	**Value**	
Atmospheric Correction	True	
Topographic Correction	True	
BRDF Correction	True	
Adjacency Effect Correction	True	
Multiple Scattering Approximation	True	
Water Vapor Correction	Null/False	Landsat only
Internal AOD Estimation	True	
Cloud Buffer (m)	300	
Shadow Buffer (m)	90	
Snow Buffer (m)	30	
FMask Cloud Threshold	0.225	
FMask Shadow Threshold	0.02	
Resolution Merge	Improphe	Sentinel-2 only
Co-Registration	Null/False	Sentinel-2 only

FORCE L2PS uses a nested parallelization strategy and settings are highly depended on each system's setup. To choose appropriate settings in this regard, the advice given by Frantz & Contributors (n.d.-c) was followed and the processing of both datasets was performed on a TerraSense node using 12 processes with 2 threads each. Using this setup, it took 1 hour and 45 minutes to process the Landsat 8 dataset, and 48 hours and 36 minutes for the Sentinel-2 dataset. A simple log file with additional information about the processing of each individual scene is written by FORCE. From these it was calculated that the actual average processing time for Landsat 8 scenes was 5 minutes and 24 seconds, whereas Sentinel-2 scenes took an average of 15 minutes and 52 seconds. This difference is expected, because of the higher spatial resolution of the Sentinel-2 data and the additional Resolution Merge processing step.

The resulting files for both datasets were written in the GeoTIFF format with band sequential interleaving (BSQ), Lempel–Ziv–Welch (LZW) compression, and internal blocks for partial image access. The size of internal blocks depends on the specifications of the overall tiling grid, which is described in Section 4.1.4. Generally, however, they are arranged as strips that are as wide as the size of each individual tile. Other GeoTIFF format settings, such as compression algorithm, are not easily changed as they are hard-coded in FORCE.

Two GeoTIFF files were created for each scene: a multi-band GeoTIFF for the Bottom Of Atmosphere (BOA) reflectance, and a single-band GeoTIFF for Quality Assurance Information (QAI). The bands of each BOA file contain data that is specific to the commonly used spectral wavelengths of optical EO sensors and are provided in a common spatial resolution. Metadata, including data that is specific to FORCE, was written into each file automatically during processing. Furthermore, to simplify the usage of data from multiple sensors the mapping of the internal bands was homogenized, which is shown in Table A-1. Additionally, Table A-2 provides more details about the information contained in the QAI files.

4.1.3 SAR Satellite Data

As mentioned in Section 4.1.1, ascending and descending datasets for the Sentinel-1 satellites were provided by the Department of Earth Observation and already processed to an ARD format. Through a Bash script that was also supplied for each scene, the individual steps of the original processing workflow could be retraced. Instead of the software SNAP, which is used by the workflow integrated in ARDCube, here, the proprietary software GAMMA was used with pyroSAR to process the datasets. In both cases, however, radiometrically terrain-corrected gamma nought backscatter data in accordance with the workflows presented by Truckenbrodt, Freemantle, et al. (2019) are produced. The exact type and spatial resolution of the DEM used for processing could not be identified through the Bash scripts but were confirmed to have been sourced from SRTM 1 arcsecond (~30 m resolution) data. In total, 1494 scenes with a size of 460 GB, and 1218 scenes with a size of 404 GB for the ascending and descending datasets, respectively, were provided.

To assure that the SAR datasets are in the same geographic projection and tiling grid as the optical datasets, the post-processing steps described in Section 3.2.3 were applied using the *process_ard* module. As a result of using the auxiliary FORCE module *force-cube*, some specifications related to the produced GeoTIFF file format are the same as described for the optical datasets, including compression algorithm, internal block size and BSQ encoding. The resulting scenes consist of two single-band

GeoTIFF files, each containing the data specific to one of the polarizations (VV, VH). General metadata was provided as an additional XML file for each scene, and no additional, FORCE-specific metadata was written into the files during the post-processing steps.

4.1.4 Projection & Tiling

Applying an appropriate geographic projection and tiling structure to ARD products is an important aspect to consider in the context of EODCs and subsequent time-series analysis of the data. FORCE has implemented a data cube structure and file organization, which is presented in more detail by Frantz (2019, pp. 2–3). The key elements of this concept are that all ARD products are reprojected into a common CRS and organized in a grid system as non-overlapping tiles. As such, the following terminology has been defined by Frantz (2019):

- "Grid": the spatial subdivision of the land surface in the target CRS.
- "Tile": a grid cell with a unique tile identifier, e.g., X0003_Y0002.
- "Chip": original images are partitioned/tiled into several chips by intersecting them with the grid.

Similar concepts have been applied to the production of Landsat ARD products (Dwyer et al., 2018), while Sentinel-2 level-1 data is already being distributed as gridded data. However, the Sentinel-2 tiling scheme uses Universal Transverse Mercator (UTM) zones, which is resulting in redundant data because of overlapping areas of the individual tiles. Further, it can cause difficulties when analyzing large areas, as each UTM zone constitutes a different projection (Roy et al., 2016).

For the processing of ARD via FORCE, multiple parameters related to projection and tiling can be specified. Two grid systems are already implemented in FORCE as default options with predefined parameters: EQUI7, which consists of seven equi-distant, continental projections, and GLANCE7 with seven equal-area, continental projections. The latter was applied to all datasets of the TDC as part of the ARD processing described in Sections 4.1.2 and 4.1.3. The resulting grid system for the TDC with square tiles covering 150 by 150 km each is shown in 4.2.

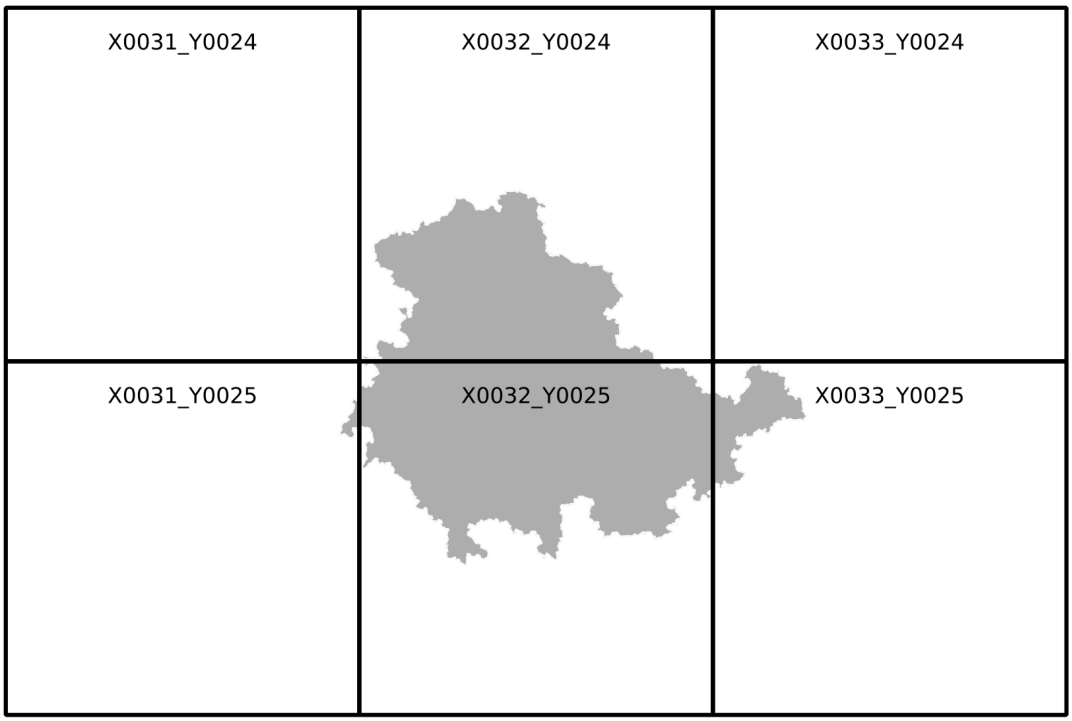

Figure 4.2: GLANCE7 EU grid over the spatial extent of the Free State of Thuringia (derived from Geoportal-Th (2021a) data). Each grid cell constitutes a non-overlapping tile covering 150 by 150 km and can be identified with a unique tile identifier.

The GLANCE7 grid was developed as part of the NASA MEaSUREs project GLobal LANd Cover and Estimation (GLANCE) designed by Boston University (Friedl et al., n.d.) and is based on the EQUI7 grid system proposed by Bauer-Marschallinger et al. (2014). It uses Lambert Azimuthal Equal Area projections to minimize distortion for each of the seven continents and ensures that areas in an ARD product are in proportion to the actual areas on the Earth's surface. More details about each continental grid is provided by Holden (n.d.). A similar equal-area projection has also been used by Dwyer et al. (2018) for a Landsat ARD product.

4.1.5 Indexing

With the *prepare_odc* module of ARDCube, the final implementation step for the TDC was performed. Hereby, ODC dataset documents were generated automatically for each dataset, and more accurately for each individual image chip after the aforementioned tiling was applied. Ultimately, a total of 405 YAML files were generated for the Landsat 8 dataset, 1102 for the Sentinel-2 dataset, as well as 1580 for the ascending and 2349 for the descending Sentinel-1 datasets. This also includes

a number of documents for very small image chips that were produced due to the tiling (e.g., tile *X0031_Y0025* in Figure 4.2).

The generated YAML files were then indexed into the ODC database using command-line functions provided by the ODC Python package. In case of the Sentinel-1 ascending and descending datasets, all files are stored in the same directory structure. Both datasets were, however, indexed as two separate products by distinguishing between their orbit directions. Ultimately, this results in an easier usage of the data, as both datasets can still be combined in an analysis when needed.

4.2 Use Cases

After completing the implementation of the TDC on TerraSense, the usability was evaluated through computations encompassing the entire spatial and temporal extent of the TDC, as well as a smaller scale time-series analysis. The former is described in Section 4.2.1 and the latter in Section 4.2.2.

As mentioned in Section 3.2.1 the Conda *user* environment facilitates usage of the TDC. As a prerequisite, the containerized PostgreSQL database needs to run as a background process in order for the ODC Python package to access the indexed data. Furthermore, a JupyterLab server was started on TerraSense to provide an interactive working environment, which could then be accessed on an external system via a Secure Shell Protocol (SSH) tunnel that simply forwards the necessary port used by the JupyterLab server.

All files related to the results presented in this section are also available in a public GitHub repository (https://github.com/maawoo/TDC_use).

4.2.1 Per-pixel Computations

4.2.1.1 Concept

To identify any problems and possible bottlenecks in terms of disk and memory bandwidth on one hand, and to get a better understanding of the spatio-temporal characteristics of the datasets on the other, per-pixel computations were performed. Hereby, for each individual pixel of the entire spatial extent of the TDC, the sum of valid (SAR datasets) and clear-sky (optical datasets) observations are calculated by considering each pixel's time-series information.

The calculation for the SAR datasets is rather straightforward, as only no data values need to be excluded or masked to get the sum of valid observations and only one of the polarization bands need to be considered as the coverage of valid data is expected to be the same.

For optical datasets, on the other hand, an appropriate clear-sky mask needs to be created from the information provided by the QAI band. In this case, the masking tool of the ODC core Python package was used to create a Boolean mask from the QAI flag values listed in Table 4.2. The values are combined in a logical *AND* fashion, which means that pixels are only set to *True* if all conditions apply. The result is a Boolean array for the entire dataset, which is then used to calculate the sum of clear-sky observations.

Table 4.2: Quality Assurance Information (QAI) flags used to create a Boolean mask for the per-pixel computation of the optical datasets (Landsat 8, Sentinel-2). A comprehensive list of QAI flags can be found in Table A-2

QAI flag	**Value**
Valid data	'valid'
Cloud state	'clear'
Cloud shadow	'no' / False

4.2.1.2 Performance Considerations

Some performance aspects need to be considered before large computations are performed. As mentioned in Section 3.1.3 the Python package Dask is handling the parallelization of computations, and more specifically the distributed scheduler of Dask is used, as, amongst others, it provides access to a diagnostic dashboard and performance reports (Anaconda, Inc. & Contributors, n.d.-b). Two aspects in particular were evaluated by using the aforementioned per-pixel computation on a spatial subset of the Sentinel-1 ascending dataset: array chunk sizes and multi-threading.

As described by Rocklin (2015), Dask uses NumPy-like arrays and blocked array algorithms internally. It can handle large computational problems efficiently by breaking up an array into smaller chunks, performing a computation per chunk and then aggregating all intermediate results. The arrangement (e.g., per dimension) and size of array chunks can affect performance and also depends on the algorithm used (Anaconda, Inc. & Contributors, n.d.-a). In case of the per-pixel computation, for

example, only the spatial dimensions need to be chunked as the algorithm requires all values along the temporal dimension.

A simple test was performed by varying the chunk size as shown in Figure 4.3. The number of tasks that are needed to ultimately come to the same computational result is also shown for both arrays. As a result of quadrupling the chunk size from 500 by 500 (A) to 1000 by 1000 (B) pixels in the spatial dimensions, the number of tasks needed decreases significantly, which furthermore decreases the total duration of the computation from 449 seconds to 196 seconds.

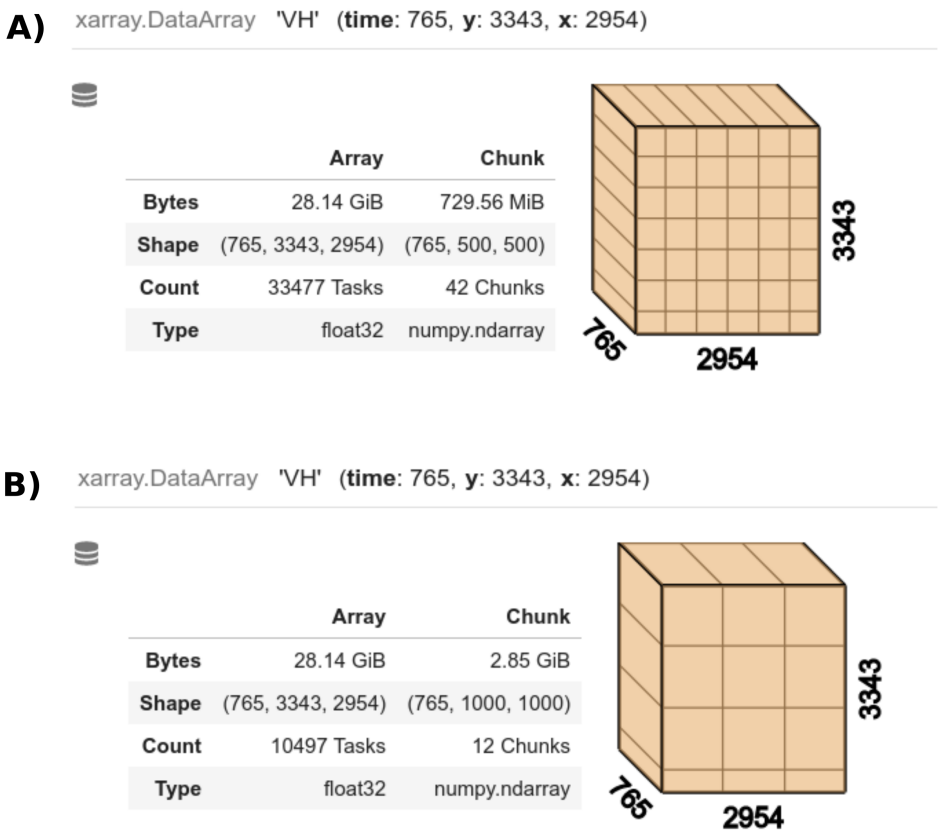

Figure 4.3: Variation of the Dask chunk size for the same dataset and performed computation. The chunk size is adjusted for the spatial dimensions (x, y), while each chunk covers the entire extent of the available time dimension. An increase of chunk size from 500 by 500 (A) to 1000 by 1000 pixels (B) decreases the number of tasks needed to perform the computation and in this case also its duration.

The second test concerns multi-threading, which is known as the ability of a single processor to follow multiple streams of execution concurrently (Nemirovsky & Tullsen, 2013, p. 1). Processors are also called *workers* in some cases, such as Dask, and streams are commonly known as *threads*. For the multi-threading test only the number of workers and threads per worker was varied, while the computation, data subset and chunk sizes remained identical. The diagnostic dashboard and reports

provided by Dask can visualize the stream of individual tasks performed by each thread and thereby reveal inefficient use of computing resources.

The task streams of 4 workers and 6 threads per worker (A), as well as 1 worker and 24 threads (B) are shown in Figure 4.4. The coloration of tasks reveals that a certain amount of communication is needed when multiple workers are utilized (A). Additionally, the computation seems to happen in a sequential order, i.e., pixel values are loaded first and the aggregation along the time dimension and per chunk is being done at the end. When a single worker has access to all available threads (B), on the other hand, a more even distribution of tasks is apparent. Furthermore, the use of allocated computing resources is more efficient and a continuous usage is facilitated. This is also reflected in a faster computation time of 196 seconds as opposed to 270 seconds in case of the former.

Figure 4.4: Variation of Dask multi-threading parameters for the same dataset and performed computation. In case of the data type and computation used here, the change from 4 workers with 6 threads each (A) to 1 worker managing all 24 threads (B), results in a more efficient and faster computation. Further information on Dask performance reports used here, can be found in Anaconda, Inc. & Contributors (n.d.-b).

4.2.1.3 Results

Based on the performance considerations tested, the actual per-pixel observations were calculated for each dataset. The results for the Sentinel-1 descending and Sentinel-2 datasets are shown in Figure 4.5 and 4.6, respectively. Similar figures are available in Appendix B for the Sentinel-1 ascending (Figure B-1) and Landsat 8 (Figure B-2) datasets.

On a larger spatial scale the number of valid observations is mostly affected by the orbits of the initial level-1 datasets. This can be observed in all cases, with some regions of fewer observations overlapping between the different datasets. Additionally, the original overlapping UTM grid of the Sentinel-2 dataset can be identified due to a slightly higher number of observations along its edges (see Appendix A, Figure A-1 for comparison).

As expected, the Sentinel-1 datasets show a rather homogenous distribution of values. However, a closer look reveals clusters of pixels with lower numbers of valid observations in comparison to surrounding areas. Most clusters appear in urban areas, as highlighted in Figure 4.5 for the state capital Erfurt, and might be related to processing artifacts due to high backscatter values.

Smaller-scale patterns are apparent in both optical datasets, due to the cloud and cloud shadow mask used for the computation. Some particular patterns correspond to false positive cloud and cloud shadow detections of the modified FMask algorithm that is used during ARD processing (Frantz et al., 2015, 2018). Examples are highlighted as A and B in Figure 4.6. Various points located in urban or industrial areas are repeatedly flagged as cloud covered, which appear as circular areas of fewer observations because of the 300 m buffer chosen during processing (see Table 4.1) (A). In other areas the extent of water bodies is outlined due to false positive cloud shadow detections (B).

Other, larger-scale patterns seem to show a natural variation due to topography. As highlighted by Figure 4.6 C, the number of clear-sky observations in the higher elevated Thuringian forest is noticeably lower than in the Thuringian basin located to the northeast (see Figure 4.1 for comparison).

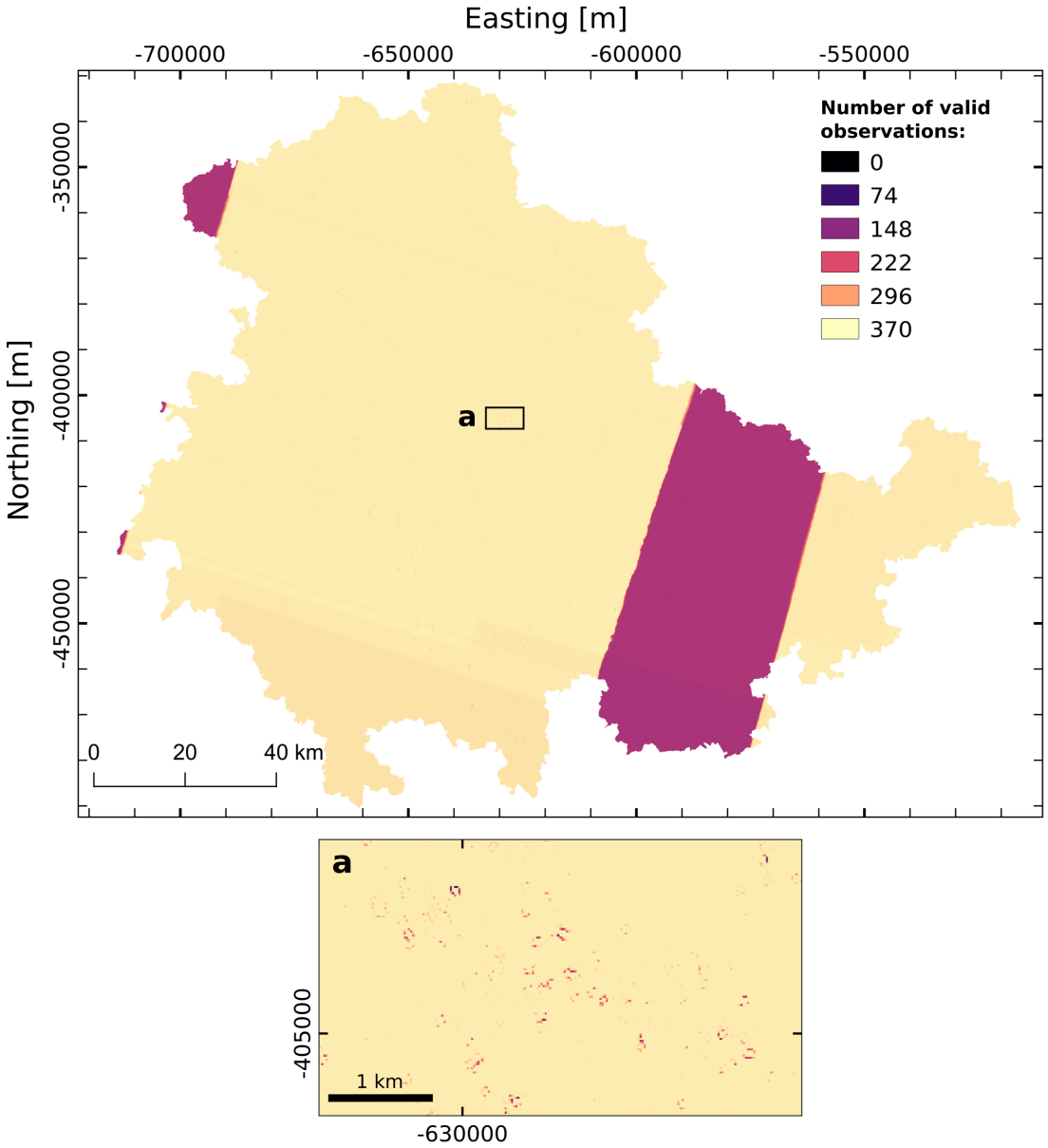

Figure 4.5: Number of valid observations between 2017-01-01 and 2019-12-31 for each available pixel of the Sentinel-1A/B descending dataset. An area of the state captial Erfurt is highlighted (a), which shows clusters of pixels with a reduced number of valid observations in comparison to surrounding areas.

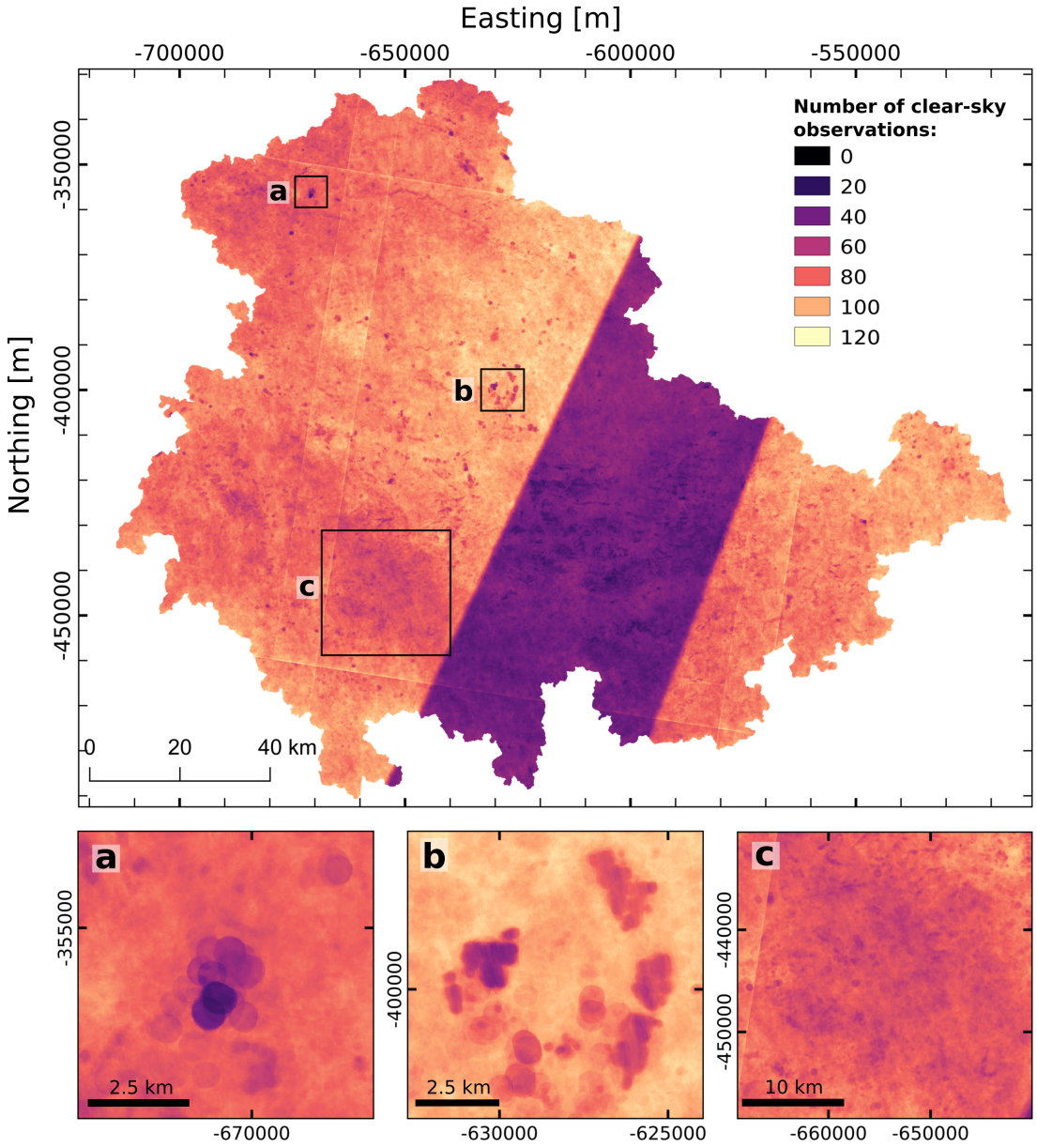

Figure 4.6: Number of clear-sky observations between 2017-01-01 and 2019-12-31 for each available pixel of the Sentinel-2A/B dataset. Various areas are highlighted: (a) shows repeated false positive cloud detections that might appear in urban or industrial areas (Frantz et al., 2018); (b) shows repeated false positive cloud shadow detections that might appear over water bodies (Frantz et al., 2015); (c) is highlighting an area of the Thuringian forest with a reduced number of clear-sky observations, which is likely related to the topography of the region (see Figure 4.1 for comparison).

4.2.2 Roda Forest Analysis

4.2.2.1 Study Area & Motivation

A further assessment of the usability of the TDC was performed through a time-series analysis that incorporates all available datasets. For this analysis the Roda forest was chosen, which is located to the southeast of the city of Jena (see Figure 4.1), and primarily contains coniferous, evergreen trees (C. Thiel & Schmullius, 2016). Figure 4.7 shows the area in more detail with a forest cover layer from an official survey overlaid (Geoportal-Th, 2021a).

The motivation behind this analysis lies in the severe summer drought that Central European forests experienced in 2018. The drought was classified as climatically even more extreme than the millennial drought of 2003 and resulted in a significant increase of drought-induced tree mortality, as well as drought-legacy effects in 2019 (Schuldt et al., 2020). The temporal extent of the current implementation of the TDC is suitable to compare the years for which most of the drought effects are expected to be observed (2018 and 2019) to a reference year (2017).

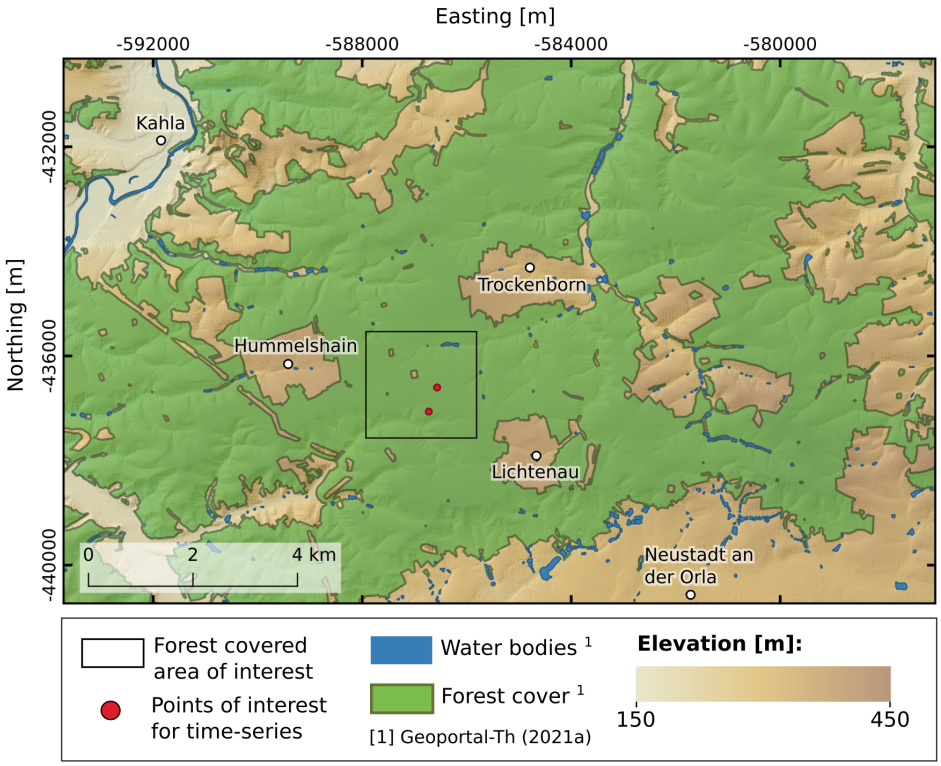

Figure 4.7: Roda forest between the cities of Kahla in the northwest and Neustadt an der Orla in the southeast. A forest covered area of interest is highlighted, which is relevant for the subsequent analysis. The map is projected in the GLANCE7 EU grid (Holden, n.d.) with corresponding easting and northing coordinates.

4.2.2.2 Methodology

For the first part of the analysis, the Normalized Difference Vegetation Index (NDVI) (Rouse et al., 1974) is calculated for all clear-sky observations of the Landsat 8 and Sentinel-2 datasets. The NDVI is a commonly used index to monitor vegetation health and is calculated using the following spectral bands:

$$NDVI = \frac{NIR - RED}{NIR + RED} \tag{4.1}$$

The Landsat 8 data is resampled using the Nearest Neighbor interpolation to the same 10 m pixel spacing as the Sentinel-2 dataset, thus creating an array of the same size and facilitating the merging of both arrays into a single NDVI dataset. As both datasets have slightly different acquisition times, no individual time steps get merged and the full temporal resolution is thereby retained.

The merged NDVI dataset is used to calculate temporal aggregates for the summer periods (June, July and August) of each available year. Hereby, for each pixel the median value per summer period is calculated. The resulting aggregated rasters can then be compared, e.g., by calculating the difference between the rasters for the summer periods 2018 and 2019 to the raster of the reference year 2017. This enables a visual assessment of possible drought effects in the region of interest. A similar raster showing the described difference between summer periods is also calculated for the Sentinel-1 ascending dataset. Here, the cross-polarized (VH) backscatter data is used, as it generally highlights volume scattering, which typically occures in forest canopies.

For the second part of the analysis the aforementioned rasters are inspected to identify forest areas that show a significant decrease in NDVI and VH backscatter values, suggesting drought-related degradation. A pixel of apparent degradation, and one located in a more stable area are then selected to visualize the time-series of all available dataset values.

4.2.2.3 Results

Figure 4.8 shows the results of the first part of the analysis. The resulting rasters are visualizing the cumulative difference (i.e., 2018 and 2019 combined) of median NDVI (A) and VH backscatter (B) relative to the summer period of 2017 and show very similar patterns overall. Some extensive areas, such as in the northwest corner, show a significant decrease in both NDVI and VH backscatter. However, as can be

seen in Figure 4.7, these areas are usually not covered by forest but most likely used for agriculture and are therefore not in the interest of this study.

A particular region is highlighted for both rasters (a = NDVI; b = backscatter), which shows a forested area in the central part of the region of interest. Here, patches of apparent degradation can be observed and two pixels were selected for the visualization of the time-series, which are shown in Figure 4.9. Furthermore, an interesting feature is only visible in the highlighted area of the calculated backscatter difference (b), where the patches of apparent degradation seem to be accompanied by patches of increased values.

The time-series plots A and B in Figure 4.9 (points P1 and P2 in Figure 4.8, respectively) confirm the reasoning behind selecting these particular points and show different developments of the values over time. While all datasets visualized for point P2 seem to be rather stable throughout the observed timeframe, a significant decrease of NDVI values and a noticeable decrease of VH backscatter values can be observed for point P1 around May 2018. It should be noted that outlier values are likely to still be present in the NDVI time-series, especially for winter months. A seasonal variation of the backscatter signal can be observed in plot B, which is particularly apparent for the descending dataset. No seasonality is apparent in the NDVI time-series, on the other hand.

Additional figures can be found in Appendix B. Figure B-3 shows the calculated NDVI difference for the summer periods 2018 and 2019 in reference to 2017 as separate plots. Figure B-4 is the equivalent of Figure 4.8 (B) for the descending Sentinel-1 dataset. Lastly, Figures B-5 and B-6 show individual plots of median NDVI and backscatter for each summer period.

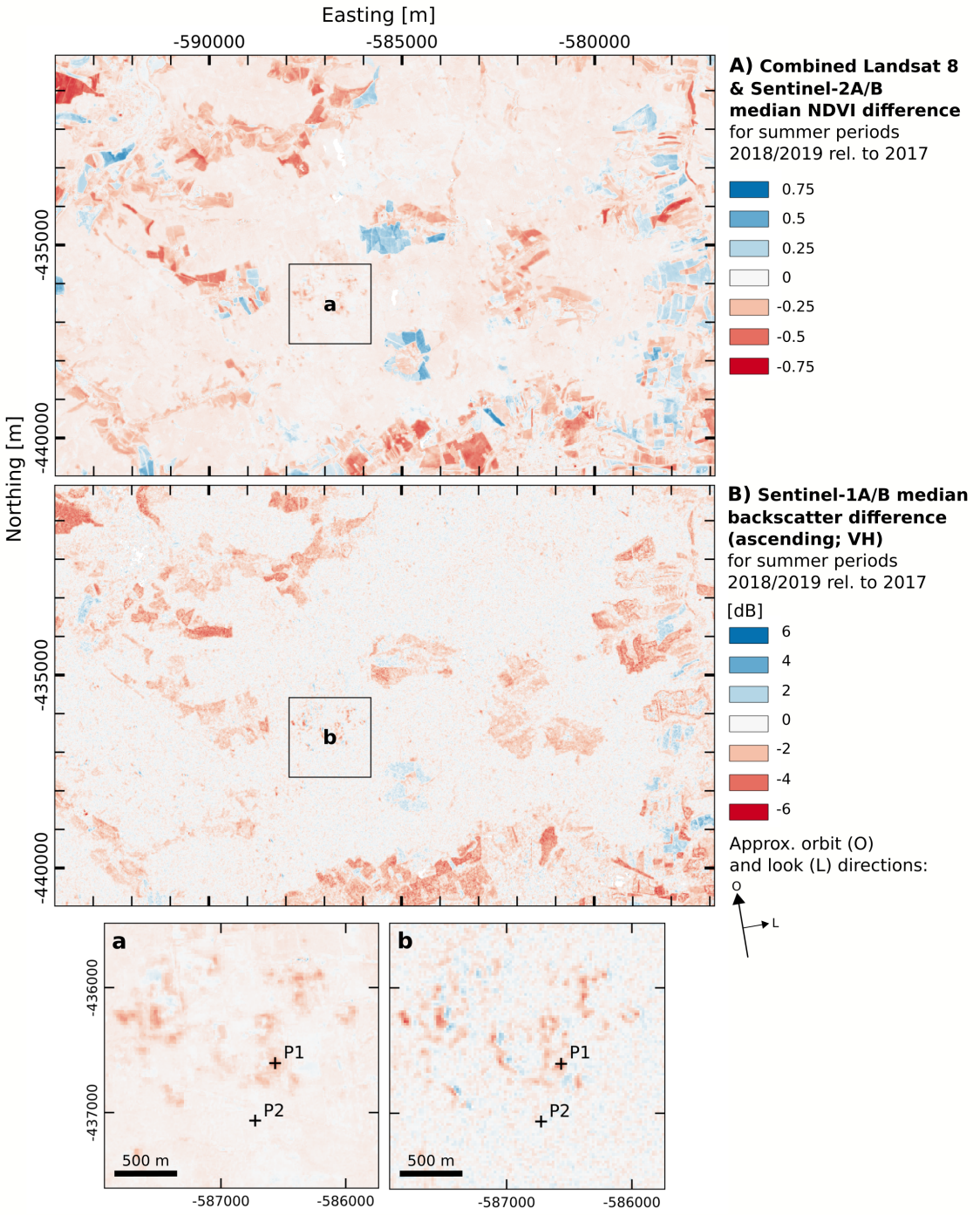

Figure 4.8: Difference of the median NDVI (Normalized Difference Vegetation Index) derived from the optical datasets (A) and SAR backscatter (ascending orbit; VH polarization) (B) for the summer periods (June, July, August) 2018 and 2019 relative to 2017. A centrally located area is highlighted for both rasters, where two points (P1, P2) were selected for the time-series visualization (see Figure 4.9). The maps are projected in the GLANCE7 EU grid (Holden, n.d.) with corresponding easting and northing coordinates.

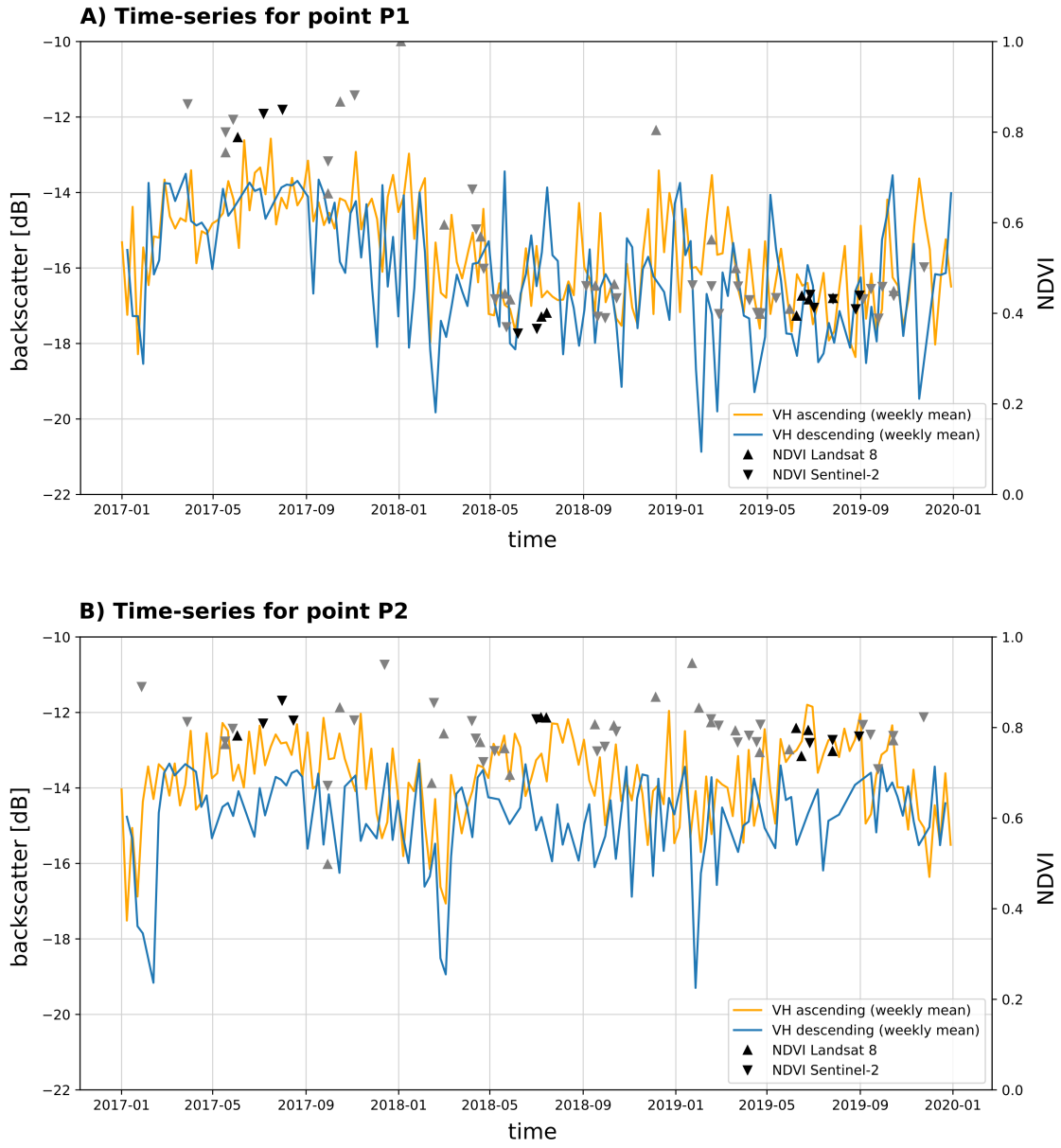

Figure 4.9: Time-series plots for the points selected in Figure 4.8. Point P1 shows a decrease in both NDVI (Normalized Difference Vegetation Index) and VH backscatter values after May 2018, whereas the values for point P2 appear to be stable over the available timespan. The backscatter values were slightly smoothed by calculating the weekly mean. NDVI values that were used to calculate the median for each summer period (see Figure B-5) are highlighted.

5 Discussion

The following discussion is divided into three sections. Section 5.1 discusses the results of the performed use cases and the usability of the TDC, which has been assessed thereby. Several possible improvements that are readily available and regard the integrated datasets are also presented. Section 5.2 focuses on technical aspects of the implementation that are related to ARD in general and the Open Data Cube. Finally, Section 5.3 provides an outlook on how the ARDCube tool could be extended, as well as the possible usage of the TDC at the Department of Earth Observation.

5.1 Usability Assessment

Per-pixel computations

By carrying out the use cases described in Section 4.2 the usability of the initial implementation of the TDC could be assessed and possible improvements identified. The computation of per-pixel observations were a particular important aspect of the assessment. Even though the calculation of the sum of valid and clear-sky observations is rather simple, the underlying logistics of such a computation is not. This is especially the case if a very large volume of data is involved that, furthermore, is heterogeneous in nature (i.a., variability in spatial and temporal coverage). Various factors can have an influence on the performance and might only be apparent or lead to problems when a computation is performed at scale, as opposed to only using a small subset of the data.

Fortunately, no significant issues were observed during the per-pixel computations of the TDC. In addition, some important aspects could be tested beforehand, which are related to the parallel computing library Dask. No universal set of parameters exists that is always resulting in optimal performance. As the tests in Section 4.2.1.2 have demonstrated, however, the available resources provided by Dask can easily be used to diagnose problems and adjust parameters accordingly. The visualization of

diagnostics and insights as an interactive dashboard is reducing abstractness and the sense of working with a black box.

The results of computing valid observations per pixel for the entire extent of the TDC (Figures 4.5, 4.6, B-1 and B-2) provide valuable information about the temporal and spatial characteristics of the datasets. On one hand, inappropriate parameterization during the processing of ARD can be identified, such as a too large buffer for cloud, cloud shadow and snow detection. Optimal parameters should of course be determined beforehand by first processing a smaller subset of the dataset. However, a visualization of the end product can reveal inconsistencies in the underlying data that might otherwise remain unnoticed. Furthermore, it can be used as an additional source of information when performing time-series analysis. An area of interest can, for example, be located in a region of significantly fewer observations due to the orbit paths of the EO satellites. In addition, valid data points might be removed unintentionally when applying masks derived from the QAI band, as the algorithm used during processing can include false-positive detections (Frantz et al., 2015, 2018). It is important to consider these aspects before conducting an analysis or at least being aware of possible impacts. Having access to this information prior to the analysis or being able to easily generate and visualize it for a particular study area can be very beneficial.

Roda forest analysis

The study area of the Roda forest is located in a region of fewer observations for the Sentinel-2 and Sentinel-1 descending datasets. Nevertheless, the amount of data points were sufficient to conduct the time-series analysis in the form that was intended and the potential of using the TDC could successfully be demonstrated.

As already described in Section 4.2.2.3, the highlighted forest area in Figure 4.8 contains patches that seem to have degraded over the observed timespan. Some of these patches are accompanied by increased VH backscatter values to the northeast, which correlates with the look direction of the ascending orbit pass of the Sentinel-1 satellites. This effect could be caused by the clearing of forest areas, as the backscatter increase might be due to a double-bounce mechanism along the newly created forest edge (Villard & Borderies, 2007). An opposite shadow effect can sometimes be observed as well, which has been utilized by Bouvet et al. (2018) to detect deforestation. However, a shadow effect does not seem to be visible in this case. The significant decrease of NDVI values for point P1 (Figure 4.9) and continuously low values thereafter, further support the assumption that at least some of these patches have not simply degraded but rather been cleared. The timing of the sharp drop-off of the NDVI time-series at the beginning of May 2018 suggests that the clearing

happened at the beginning of the 2018 drought (Schuldt et al., 2020). A reason for such a clearing could be to prevent the spread of a bark beetle infestation. Damage caused by insect infestations has developed into the main reason for logging in German forests in recent years (Destatis, 2021), hence increasing the need for monitoring solutions that use time-series information derived from EO data (e.g., Fernandez-Carrillo et al., 2020; Hollaus & Vreugdenhil, 2019).

The lack of seasonality in the NDVI time-series for point P2 (Figure 4.9) indicates that a coniferous evergreen forest is present in the area (cf. She et al., 2015). A seasonal variation of the VH backscatter time-series on the other hand is plausible as the signal is influenced by the water content of the foliage, which changes over the year according to water availability (Dubois et al., 2020). The ascending and descending signals appear to show a more pronounced division during the summer periods 2018 and 2019 in comparison to 2017. As the orbits pass over the area during different times of the day (late afternoon for ascending; early morning for descending orbit), the diurnal difference of the backscatter signals could indicate drought related water stress (Steele-Dunne et al., 2012). This effect needs to be investigated in more detail, however, to confirm or disprove this hypothesis.

Possible legacy effects of the 2018 drought can be observed in Figure B-3. Most of the forest cover in the study region (see Figure 4.7) shows slightly lower median NDVI values in 2019 in comparison to 2018 when the drought occurred. This apparent decrease of vegetation health could be due to direct or indirect lagged impacts following the year of an extreme drought (Frank et al., 2015). As with the previous hypothesis, further investigation is needed to also exclude possible flaws in the methodology used for this analysis.

Dataset considerations & improvements

Data points from the Sentinel-2 and Landsat 8 datasets have been combined to calculate the median NDVI differences seen in Figures 4.8 and B-3. This combined usage is not trivial, as both sensors cover slightly different spectral wavelengths. While effort is being made to create harmonized data products from both sensors (Claverie et al., 2018; Scheffler et al., 2020), ARD products that have been created with FORCE are not yet harmonized in terms of spectral wavelengths during processing. Therefore, this aspect needs to be considered when similar time-series analysis are conducted using the TDC and the appropriateness of aggregating data points from different datasets ultimately depends on the study objective.

Currently, the Level-1C data products of the Sentinel-2 mission have a multi-temporal geometric uncertainty of around 12 m (Gascon et al., 2017), which can have a negative influence on time-series analyses. While a geometric

refinement is currently being implemented by ESA to improve the accuracy (ESA, 2021), it is not clear if the reprocessing of past datasets is planned. A coregistration option has already been implemented into FORCE L2PS (see Figure 3.1). Rufin et al. (2020) describe the algorithm used in more detail, which ultimately leverages base images created from the Landsat 8 near-infrared band to improve the multi-temporal geometric uncertainty of the processed Sentinel-2 ARD product to an average of around 4.4 m. This processing option requires some preparation steps but is readily available in the version of FORCE utilized in the ARDCube tool. The quality of time-series can therefore be improved for future analyses performed with the TDC by reprocessing the Sentinel-2 dataset and performing the coregistration step.

The Sentinel-1 datasets were already provided in an ARD format and have been processed using an SRTM 1 arcsecond DEM. As Truckenbrodt, Freemantle, et al. (2019) describe in their work, large discrepancies can sometimes be observed between various openly available DEM options, such as SRTM. This can affect the quality of topographic normalization during processing and ultimately the time-series analysis of individual pixels. Similarly to the Sentinel-2 dataset, it might be worthwhile to reprocess the dataset to further improve the data quality in regard to time-series analysis. The LiDAR-derived DEM utilized during the processing of the optical datasets could be used after a similar quality assessment as described by Truckenbrodt, Freemantle, et al. (2019) has been performed.

Concluding remarks

In conclusion, the usability of the current implementation of the TDC has successfully been demonstrated with the performed use cases. The quality of all datasets is appropriate for time-series analysis but could be further improved with readily available processing options and ancillary data. Access to the indexed ARD products via the ODC Python API works without any issues or additional preparation steps. For example, no reprojection and resampling of the data has to be performed during loading of the data as Xarray datasets, as the data is already stored in a common CRS and the same non-overlapping tiling scheme. The utilization of a continental projection, such as GLANCE7, can be additionally advantageous in the future by facilitating interoperability with other study areas in the same region, provided that the same projection is used.

The existing ecosystem of Python packages surrounding the core packages Xarray and Dask is steadily growing and has been adopted by a variety of scientific fields and institutions (e.g., Eynard-Bontemps et al., 2019). The aspect of adoption should not be neglected in connection with open-source software projects, as it can ensure

long term support of development. Various packages in this ecosystem can pave the way to more advanced types of analyses than demonstrated here. The Dask extension dask-ml (Dask-ML Contributors, n.d.), for example, provides access to scalable machine learning by leveraging the popular Python library Scikit-Learn (Pedregosa et al., 2011). Furthermore, packages that might not be directly related to the ecosystem can easily be integrated into an analysis, such as the Roda use case. Additional information can thereby be provided, like climatic time-series data from weather stations located in the study area of interest (Gutzmann et al., n.d.).

5.2 Implementation Assessment

5.2.1 Analysis Ready Data

Processing optical and SAR data with the software components integrated in the ARDCube tool produces datasets that can directly be used in an analysis and hence be designated as ARD products. However, a formal assessment of how well these products comply with the current CARD4L specifications described in Section 2.3.1, has neither been done in the course of this work, nor by the developers of FORCE or pyroSAR. However, Truckenbrodt, Freemantle, et al. (2019) acknowledge that this is a future goal along with a relevant extension of the pyroSAR software, and at the time of writing the official CARD4L website already lists FORCE as an ARD resource (CEOS, n.d.-a).

Various aspects that are related to ARD were taken into account while developing the ARDCube tool and implementing the TDC, but were not actively adjusted or changed. For the data format of both optical and SAR datasets, for example, the current setup solely relies upon which GeoTIFF specifications are defined by FORCE for the output files. Alberti (2018) demonstrates that GeoTIFF specifications, like the compression algorithm used, can have significant impacts on read and write speeds, as well as the ratio of compression and therefore storage size. Moreover, the internal tiling of the files affects the performance when only a small part of each file is accessed, which is done repeatedly when retrieving a pixel time-series, for example. These aspects need to be considered when large volumes of EO data are supposed to be handled during an analysis. Furthermore, new raster data formats have emerged in recent times, including COG and Zarr, which provide their own set of specifications, as well as advantages and disadvantages (Yee et al., 2020). In regard to the TDC, the current approach works quite well as demonstrated with the per-pixel computations (Section 4.2.1), and additionally, the option to use the

higher-level processing system of FORCE (Frantz, 2019, pp. 9–15) is retained by not adjusting the output format after processing. Therefore, any changes of the data format need to be justified, as existing datasets and the processing workflows would need to be adjusted. There appears to be a lack of literature in regard to this topic, so an assessment where different raster data formats and their specification options are compared in the same computational setup could be valuable.

Similar to the data format, the handling of metadata needs further consideration. As mentioned in Sections 4.1.2 and 4.1.3, additional metadata is stored in each GeoTIFF file by FORCE, whereas the SAR datasets were provided with separate metadata files. At present, the ARDCube tool does not include any ancillary functionality or leverages relevant features provided by pyroSAR, for example, to organize the available metadata. Another layer of complexity is added by the ODC, which requires its own set of metadata files in the form of dataset documents (see Section 3.1.3), which currently only contain necessary information to ensure that the ODC Python API is operable. The CARD4L specifications list *machine readability* as one of the minimum requirements in terms of the handling of metadata, which is satisfied in all cases mentioned. Nevertheless, a uniform solution to organize and easily access the metadata of all datasets would be desirable. The SpatioTemporal Asset Catalog (STAC) specification (Radiant Earth Foundation & Contributors, n.d.-a) could provide a viable and standardized solution, which potentially will be implemented into the ODC in the near future (ODC Contributors, n.d.-c). It uses a similar approach as used by the ODC in the form of higher-level (*Product Definition* in ODC; *Collections* in STAC) and lower-level metadata documents (*Dataset Documents* in ODC; *Items* in STAC). The STAC specification is an open-source project that is already being supported by a large community and currently receives a lot of attention in the geospatial domain. Openly available tools are actively being developed, such as the STAC Browser (Radiant Earth Foundation & Contributors, n.d.-b), providing intuitive methods of exploring the organized metadata.

A final aspect to be addressed is the availability of auxiliary data products, which are specified by CARD4L as *Per-Pixel Metadata* (see Section **??**). The QAI band produced by FORCE for the optical datasets is an auxiliary product already used in this implementation of the TDC. It provides valuable information during analyses and can be used to filter a time-series for clear-sky observations as demonstrated in Section 4.2.1. For the SAR datasets on the other hand, no auxiliary data products have currently been created and integrated into the TDC. The Normalized Radar Backscatter PFS of CARD4L lists, amongst others, the provision of a local incident angle image as a minimum requirement. In addition, Truckenbrodt, Freemantle, et al. (2019) recommend that a map of geometrical distortion (e.g., layover and radar

shadow) should also be provided when a Sentinel-1 EODC is created. Both products can be produced during processing with pyroSAR's SNAP API and integrating them into the TDC could be of great value for future analyses and the development of new methodologies.

5.2.2 Open Data Cube

Gentemann et al. (2021) state that a paradigm shift is happening in science, as data, software, and computational resources are moving towards cloud-based solutions. However, a lot of challenges are yet to be solved and HPC systems will remain important tools in many research institutions while technologies related to this shift are increasingly being adopted (Abernathey et al., 2020). In conjunction with this trend, various cloud-based EO platforms have emerged. As described by Giuliani, Masó, et al. (2019), such platforms potentially come with their own set of drawbacks and limit the control and flexibility of users. These aspects are particularly important in regard to research departments where existing computational resources are often utilized to perform data-intensive workloads and supplementary data sources might need to be integrated into analyses. Therefore, the ODC has been an appropriate choice for the initial implementation of the TDC.

The complete setup of an operational EODC based on the ODC software library is still rather complex, as a lot of factors need to be considered and both IT and remote sensing knowledge is required. This problem has also been pointed out by Giuliani, Chatenoux, et al. (2020) and Hernández-López et al. (2021), and is limiting the adoption of this technology. A possible solution has been proposed by Giuliani, Chatenoux, et al. (2020) in the form of the Docker-based Data Cube on Demand (DCoD). This project intends to cover the entire cycle of downloading and processing EO data for a particular area of interest, as well as creating an ODC instance, with an automated and on-demand system. Being open-source with possible improvements and development of the proposed project through external contributors has been named by the authors as an additional advantage, but the DCoD has yet to be made public. The ARDCube project developed in the course of this work intends to solve the same problem as the DCoD, while focusing on the deployment on HPC systems. As mentioned in Section 3.1.4, Singularity has been selected as the containerization solution for this reason, whereas Docker usually prevails as the most popular solution. This is also the case in regard to ODC, as various Docker-based ODC projects exist, but none have yet been found that are based on Singularity. Furthermore, the current documentation of ODC is lacking in clear guidance regarding the existing Docker-based projects, which is possibly

resulting in confusion amongst ODC users and has complicated the search for an appropriate solution in regard to the TDC implementation.

It has already been mentioned in Section 5.2.1 that ODC is moving towards integrating the STAC specification. Similarly, the database backend is likely to change at some point in the future, as various proposals have been made to either replace the ODC database API (Kouzoubov et al., 2019), which is currently relying on PostgreSQL, or to add support for alternative backends (Dhar, 2021; Woodcock, 2019). However, it is yet unknown if and when these changes are implemented into the core library of ODC. A project that not only shares some of the same core Python packages as ODC but also drives their development forward, is Pangeo (Pangeo Contributors, n.d.-a). This community-driven project aims to enable big data geoscience research through an ecosystem of open-source software packages, such as the aforementioned Xarray, Dask and JupyterLab, and is already being used with large volumes of EO data as demonstrated by Kellndorfer (2021). As the ODC is still in a transformative state of development, it might be worthwhile to explore alternative solutions that can be used to organize and access the EO datasets of the TDC. The Pangeo software ecosystem is complemented by STAC related projects, such as Intake-STAC (Hamman, 2020) and StackSTAC (Joseph & Contributors, n.d.), and might already facilitate the development of a more lightweight architecture that is similar to what is being envisaged for the future of the ODC. However, a more in-depth comparison would be needed to identify advantages and disadvantages, as well as any technical limitations that could be present at the current stage of development.

Coetzee et al. (2020) have discussed the importance of communities that ultimately drive open-source projects forward in their development and that sustaining such a community can be challenging. This aspect is an important challenge for the future development of the ODC. Even though it is one of the pioneering projects regarding EODCs and has been deployed in various countries and regions worldwide, it still lacks an appropriate space for the community of users to engage sustainably. However, this challenge is likely being addressed, since the ODC has recently joined OSGeo (Open Source Geospatial Foundation) as a community project (OSGeo, n.d.), and is being supported via the Open Earth Alliance community activity of GEO (Group on Earth Observations) (Gowda, n.d.).

5.3 Outlook

The ARDCube tool developed in the course of this work successfully facilitated the implementation of the TDC on the TerraSense HPC system. While the usage still requires a certain degree of IT and remote sensing knowledge, it can ease the process of creating an EODC for a region of interest and could potentially be used by other individuals or research departments. For this reason, the aim is to further develop the ARDCube tool and extend its functionality.

A particular area of improvement that might be of great potential is reproducibility. Abernathey et al. (2020) have stated that the "reproducibility of data science projects requires open access to at least three elements: the code, the software environment, and the data." (p. 3). They propose the concept of cloud-native data repositories, which could enable the reproducibility of scientific results that depend on large volumes of data and computational resources. However, various challenges remain that limit adoption of cloud computing in scientific research (e.g., funding). One of the few possible options for reproducibility in such cases is the recreation of the necessary software environment and datasets. Recreating a software environment and reprocessing data to the specific format necessary to reproduce scientific results comes with its own challenges, but can be eased by using containerization, for example. The ARDCube tool already provides the necessary foundation for reproducibility by managing multiple containerized software components and their parameterization. A possible improvement of the tool could be the automatic creation of some kind of *recipe* that defines all necessary information to recreate an EODC. This includes information such as the query used to download level-1 data, processing parameters for ARD generation, and software versions and dependencies to recreate the software environment. The recipe itself could be realized in the form of a simple machine-readable file (e.g., in the JSON format) that is fed back to ARDCube, or by utilizing an existing workflow management system like Snakemake (Mölder et al., 2021).

Various improvements of the TDC are possible in the near future. Some improvements regarding the EO datasets that have already been implemented (Landsat 8, Sentinel-2A/B and Sentinel-1A/B) were suggested in the previous sections. The volume of data can be further extended by processing the data from other sensors of the Landsat archive via FORCE, which opens up the possibility to investigate time-series that span multiple decades. Other SAR datasets could be implemented as well via existing format drivers provided by pyroSAR. This would require some adjustments of the ARDCube tool, however, whereas processing of data from other Landsat

sensors is immediately possible. Furthermore, the usage of Sentinel-1 Single Look Complex data in EODCs is being explored (Eurac Research & Dares Technology, n.d.). This would be a valuable addition to the TDC and enable the development of new methodologies for time-series analysis of SAR data.

Once the volume of data has been extended temporally and in terms of additional sensors, the TDC could be kept up-to-date by automating the ARDCube workflows on TerraSense with a set of cron jobs (Vixie et al., n.d.). This could further be extended by automatically deriving information products from the TDC, which could be made accessible to the public through interactive web applications. The development of such an application could leverage existing projects of the ODC ecosystem (Gowda & Killough, 2020; ODC Contributors, n.d.-b).

Lastly, access to the TDC needs be improved to facilitate the efficient usage by multiple users and to avoid the possible misuse or accidental blocking of computational resources on TerraSense. This would especially be relevant in case working with the TDC is integrated into any existing lectures and a number of students need access simultaneously. Fortunately, solutions already exist that can also be adapted on TerraSense for the TDC, such as JupyterHub (Thomas et al., 2021).

The integration of EODCs at the Remote Sensing Department of the University of Würzburg (M. Thiel et al., n.d.) provides a possible outlook of how the TDC could be further utilized. Moreover, an increase in collaboration and sharing of knowledge between both departments could be of great potential. This might also include an effort to improve the interoperability between existing EODCs, which has been named by Giuliani, Masó, et al. (2019) as an essential challenge in order to prevent individual EODCs to become silos of information.

6 Conclusion

The Thuringian Data Cube has successfully been implemented on the HPC system of the Department of Earth Observation at the Friedrich Schiller University Jena. This initial implementation includes EO data in an analysis-ready format from the optical satellites Landsat 8 and Sentinel-2A/B, as well as the SAR satellites Sentinel-1A/B, for a three-year period.

The implementation of the TDC was facilitated by the development of the ARD-Cube tool, which enables the management of multiple workflows that are necessary to create an operational EODC. The tool covers the download of EO datasets, the processing of said datasets to ARD products organized in a non-overlapping grid system, and finally, the preparation of the ARD products to be managed and accessed through the Open Data Cube software library. Furthermore, the usage of open-source software components that are encapsulated as Singularity containers, eases the setup process, enables the usage on HPC systems, and supports the principle of open and reproducible science.

Two use cases were performed that not only helped in assessing the general usability of the TDC implementation, but also in demonstrating the potential of performing small- and large-scale analyses with the integrated ARD products, as well as available open-source Python tools. The large-scale computations of per-pixel observations were primarily intended to identify any possible issues regarding the ARD products and the technical setup in general. The results of the computations, however, also provide additional, valuable information for subsequent analyses performed with the TDC. The smaller-scale Roda forest analysis, on the other hand, was carried out to demonstrate a time-series analysis that involves all available ARD products. At the same time, the prevailing issue of drought related impacts on forest ecosystems in the region could be studied, and has opened up the possibility of follow-up analyses to further investigate various features that were observed in the results.

Both the ARDCube tool and the TDC can be further improved and extended. The ARDCube tool, for example, offers the potential to create a thoroughly reproducible workflow that includes all aspects of the creation of EODCs and any subsequent analysis. The current implementation of the TDC, on the other hand, could be optimized and extended in a variety of aspects despite already being a capable and operational system. This includes the general accessibility of the TDC on the HPC system and more specifically the extension of the ARD products in terms of temporal extent, variety of optical and SAR sensors, as well as the addition of ancillary data products. Furthermore, current advancements related to the organization and access of large volumes of EO data and its metadata, such as STAC and Pangeo, could offer viable alternatives to the current approach of using the ODC and should be further explored and compared.

In conclusion, the initial implementation of an Earth Observation Data Cube for the Free State of Thuringia successfully establishes a foundation for the efficient management and analysis of EO data that is characterized by various challenges inherent to big Earth data (e.g., volume, velocity, and variety). A thriving ecosystem of related open-source software projects facilitates the development of new and innovative analysis methods that can be performed at large scales, covering the entire spatial and temporal extents available. Valuable information can thereby be extracted from the EO data, which helps in better understanding a variety of Earth system components and can ultimately support the adaptation and mitigation of climate related impacts that will most likely continue to gain importance in the future.

Appendix

Appendix A — TDC Implementation

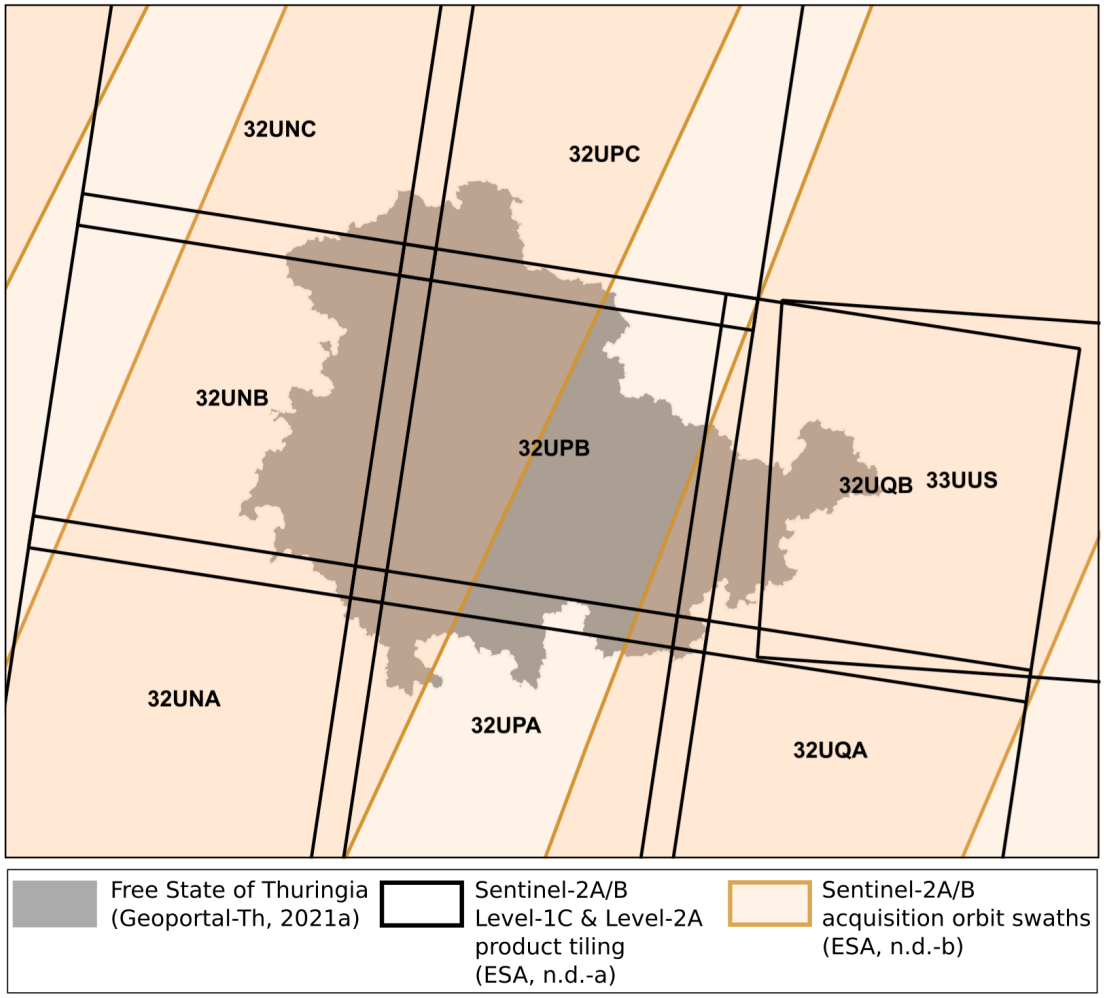

Figure A-1: Sentinel-2A/B product tiling scheme and and acquisition orbit swaths over the Free State of Thuringia.

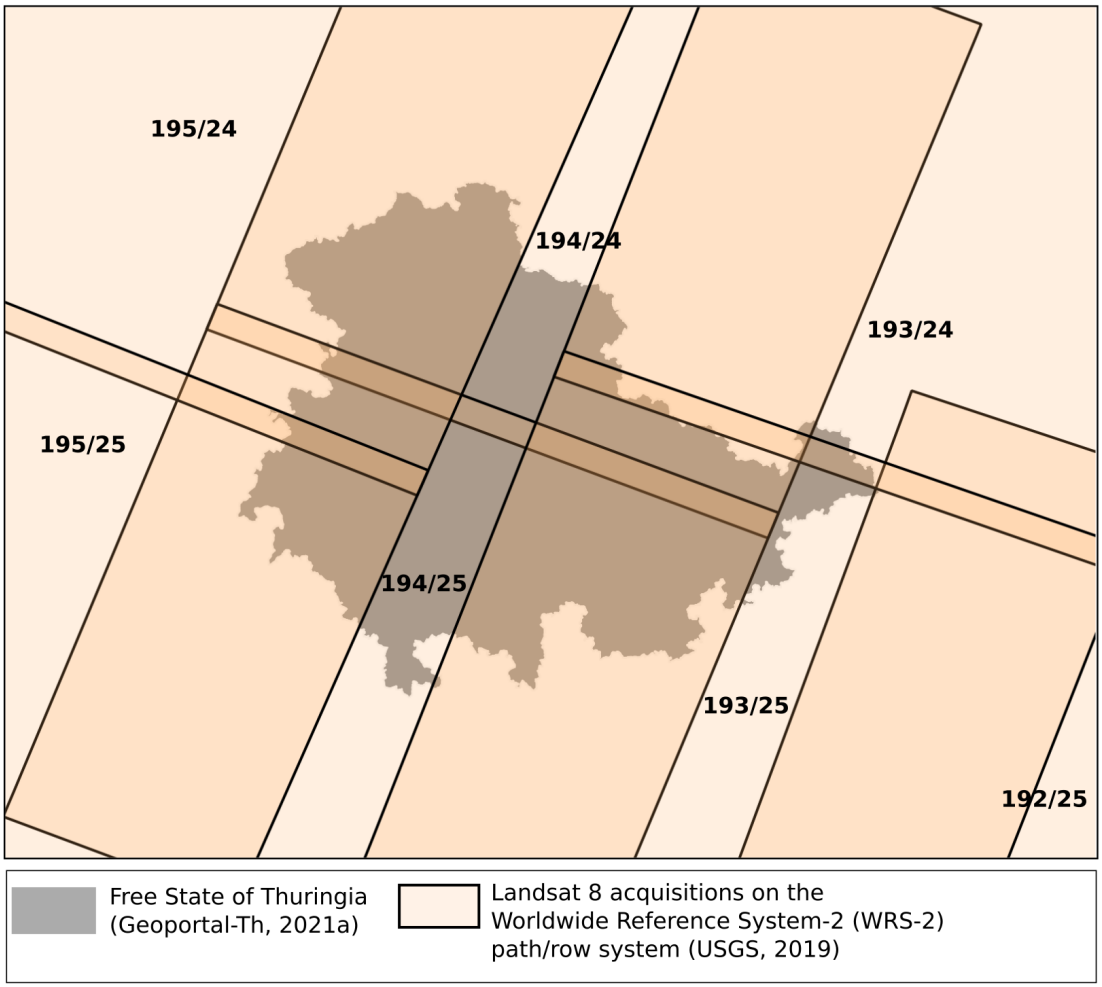

Figure A-2: Landsat 8 product tiling scheme over the Free State of Thuringia.

Figure A-3: Sentinel-1A/B example acquisition footprints for the ascending (A) and descending (B) orbits over the Free State of Thuringia.

Table A-1: FORCE level-2 output bands and mapping to original level-1 bands. Adapted from Frantz et al. (n.d.-b) and Frantz (2019).

Wavelength Designation	**FORCE Level-2 (Landsat 4-8)**	**FORCE Level-2 (Sentinel-2A/B)**	**USGS Level-1 (Landsat 4/5/7)**	**USGS Level-1 (Landsat 8)**	**ESA Level-1 (Sentinel-2A/B)**
BLUE	1	1	1	2	2
GREEN	2	2	2	3	3
RED	3	3	3	4	4
REDEDGE1	-	4	-	-	5
REDEDGE2	-	5	-	-	6
REDEDGE3	-	6	-	-	7
BROADNIR	-	7	-	-	8
NIR	4	8	4	5	8A
SWIR1	5	9	5	6	11
SWIR2	6	10	7	7	12

Table A-2: FORCE level-2 per-pixel Quality Assurance Information (QAI) description. IA = Incidence Angle; TC = Topographic Correction. Adapted from Frantz et al. (n.d.-b) and Frantz (2019).

Bit No.	**Parameter Name**	**Bit Comb.**	**Integer**	**State**
0	Valid data	0	0	valid
		1	1	no data
1-2	Cloud state	00	0	clear
		01	1	less confident cloud
		10	2	confident, opaque cloud
		11	3	cirrus
3	Cloud shadow	0	0	no
		1	1	yes
4	Snow	0	0	no
		1	1	yes
5	Water	0	0	no
		1	1	yes
6-7	Aerosol state	00	0	estimated (best quality)
		01	1	interpolated (mid quality)
		10	2	high (aerosol optical depth > 0.6)
		11	3	fill (global fallback, low quality)
8	Subzero	0	0	no
		1	1	yes
9	Saturation	0	0	no
		1	1	yes
10	High sun zenith	0	0	no
		1	1	yes (sun elevation $< 15°$)
11-12	Illumination state	00	0	good (IA $< 55°$, best quality for TC)
		01	1	medium (IA 55°-80°, good quality for TC)
		10	2	poor (IA $> 80°$, low quality for TC)
		11	3	shadow (IA $> 90°$, no TC applied)
13	Slope	0	0	no (cosine correction applied)
		1	1	yes (enhanced C-correction applied)
14	Water vapor	0	0	measured (best quality, only Sentinel-2)
		1	1	fill (scene average, only Sentinel-2)

Appendix B — TDC Use Cases

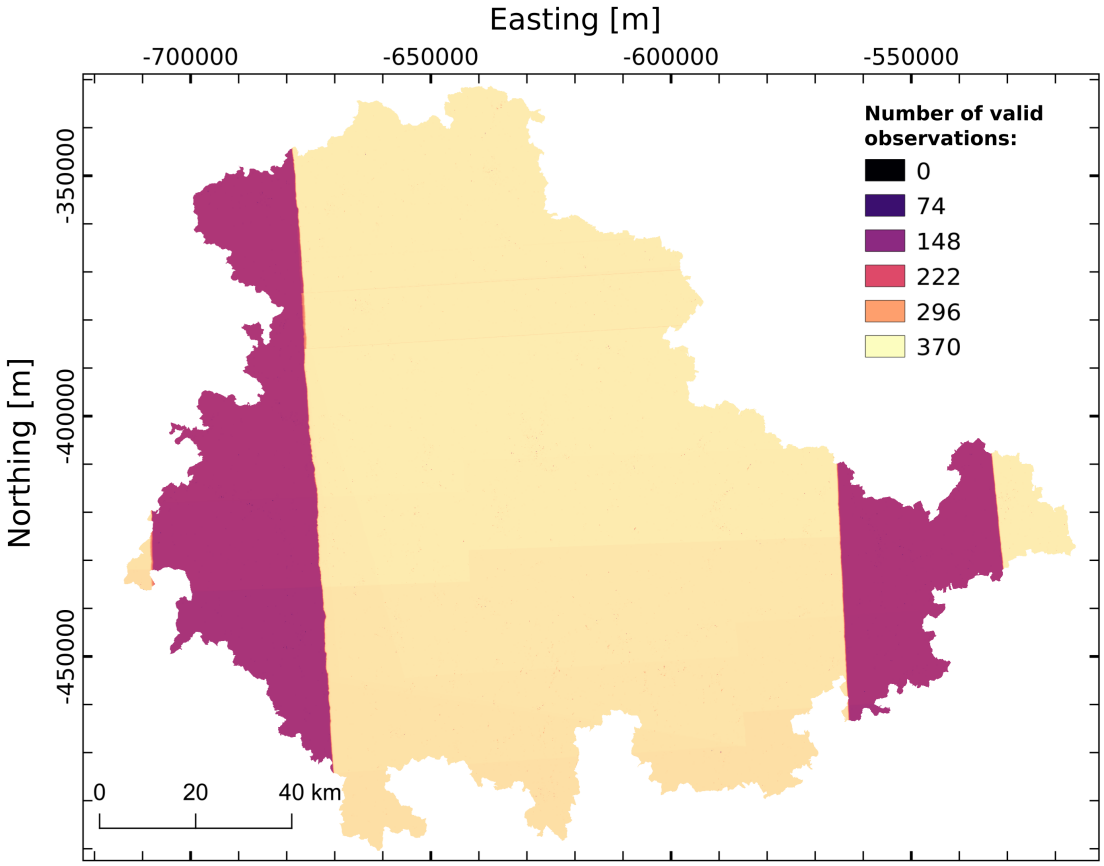

Figure B-1: Number of valid observations between 2017-01-01 and 2019-12-31 for each available pixel of the Sentinel-1A/B ascending dataset.

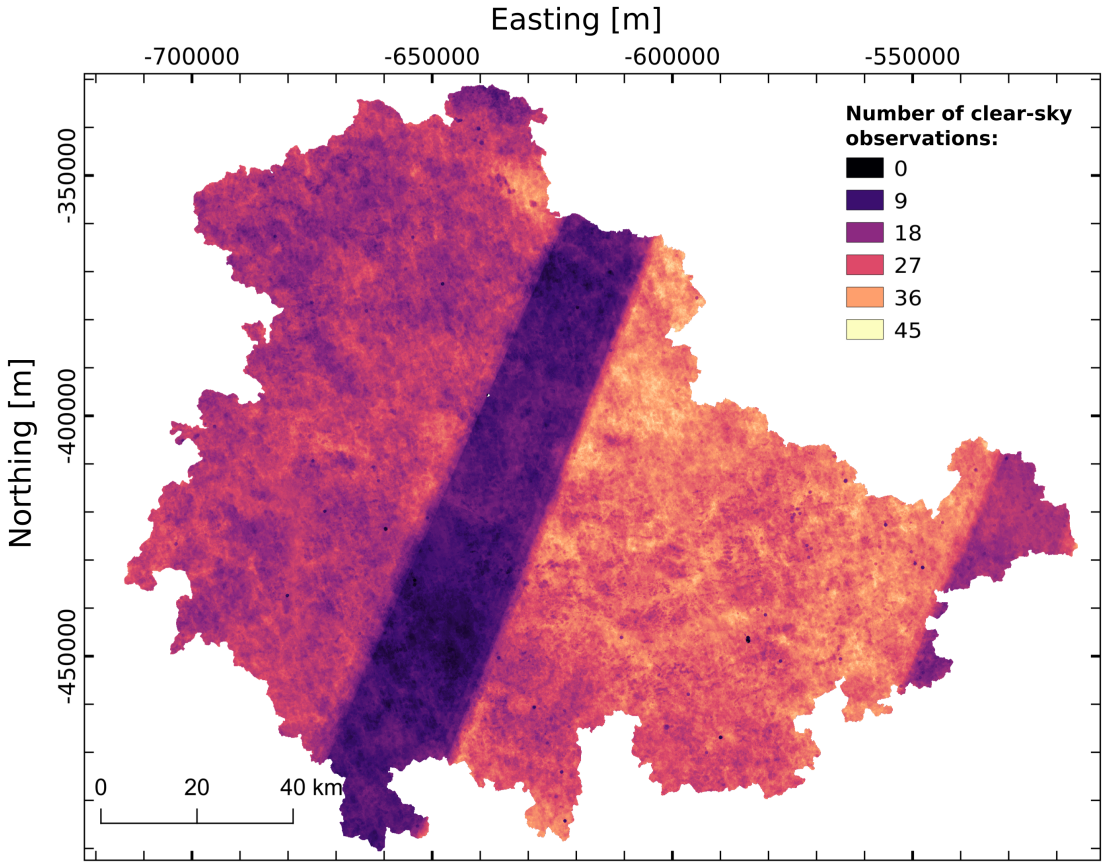

Figure B-2: Number of clear-sky observations between 2017-01-01 and 2019-12-31 for each available pixel of the Landsat 8 dataset.

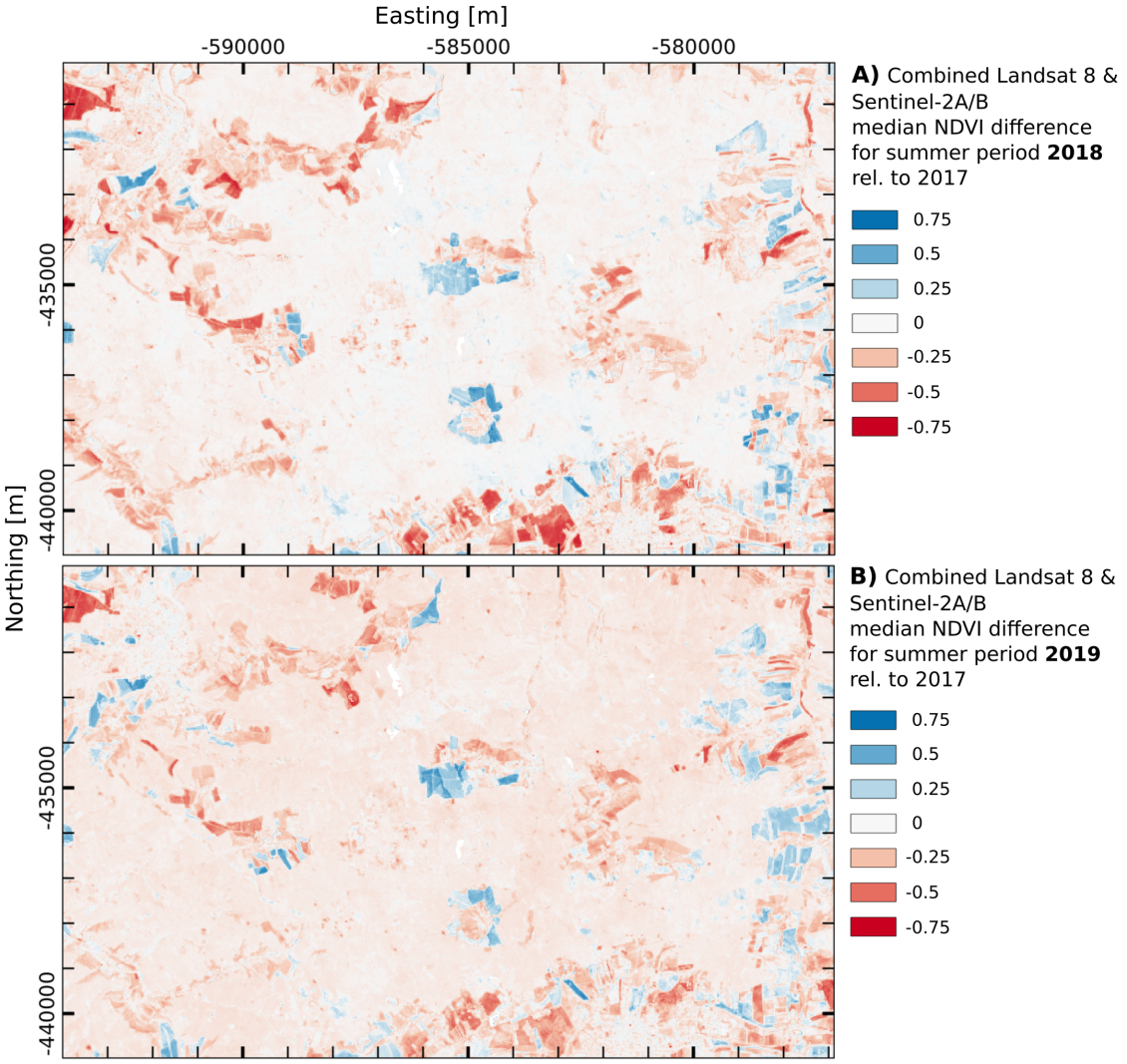

Figure B-3: Difference of the median NDVI (Normalized Difference Vegetation Index) derived from the optical datasets for the summer periods (June, July, August) 2018 (A) and 2019 (B) relative to 2017. The maps are projected in the GLANCE7 EU grid (Holden, n.d.) with corresponding easting and northing coordinates.

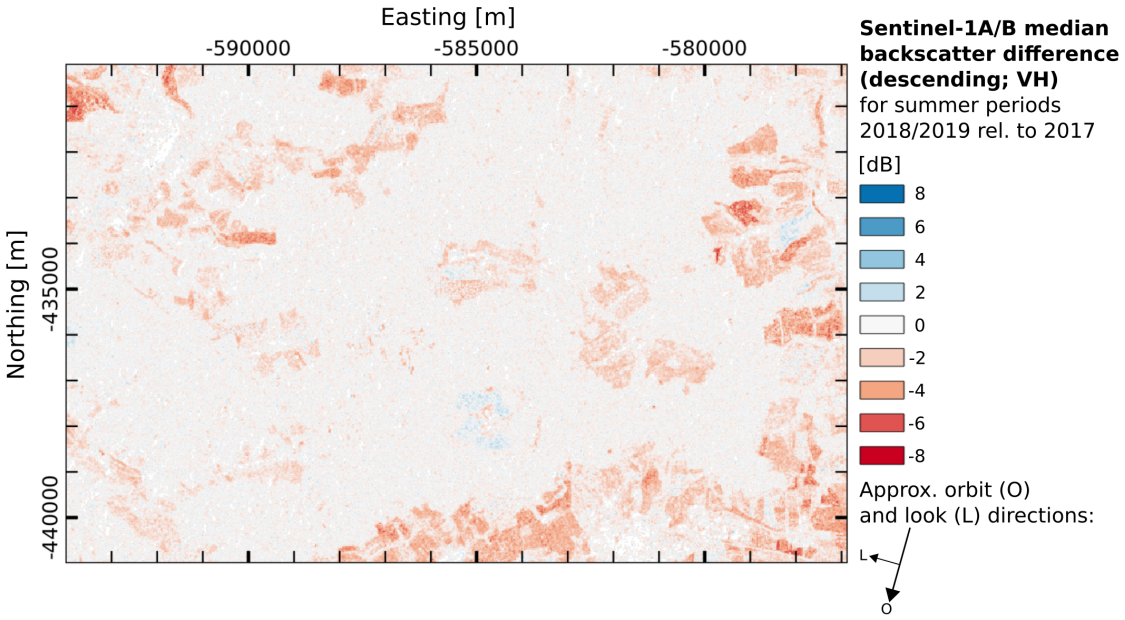

Figure B-4: Difference of the median SAR backscatter (descending orbit; VH polarization) for the summer periods (June, July, August) 2018 and 2019 relative to 2017. The maps are projected in the GLANCE7 EU grid (Holden, n.d.) with corresponding easting and northing coordinates.

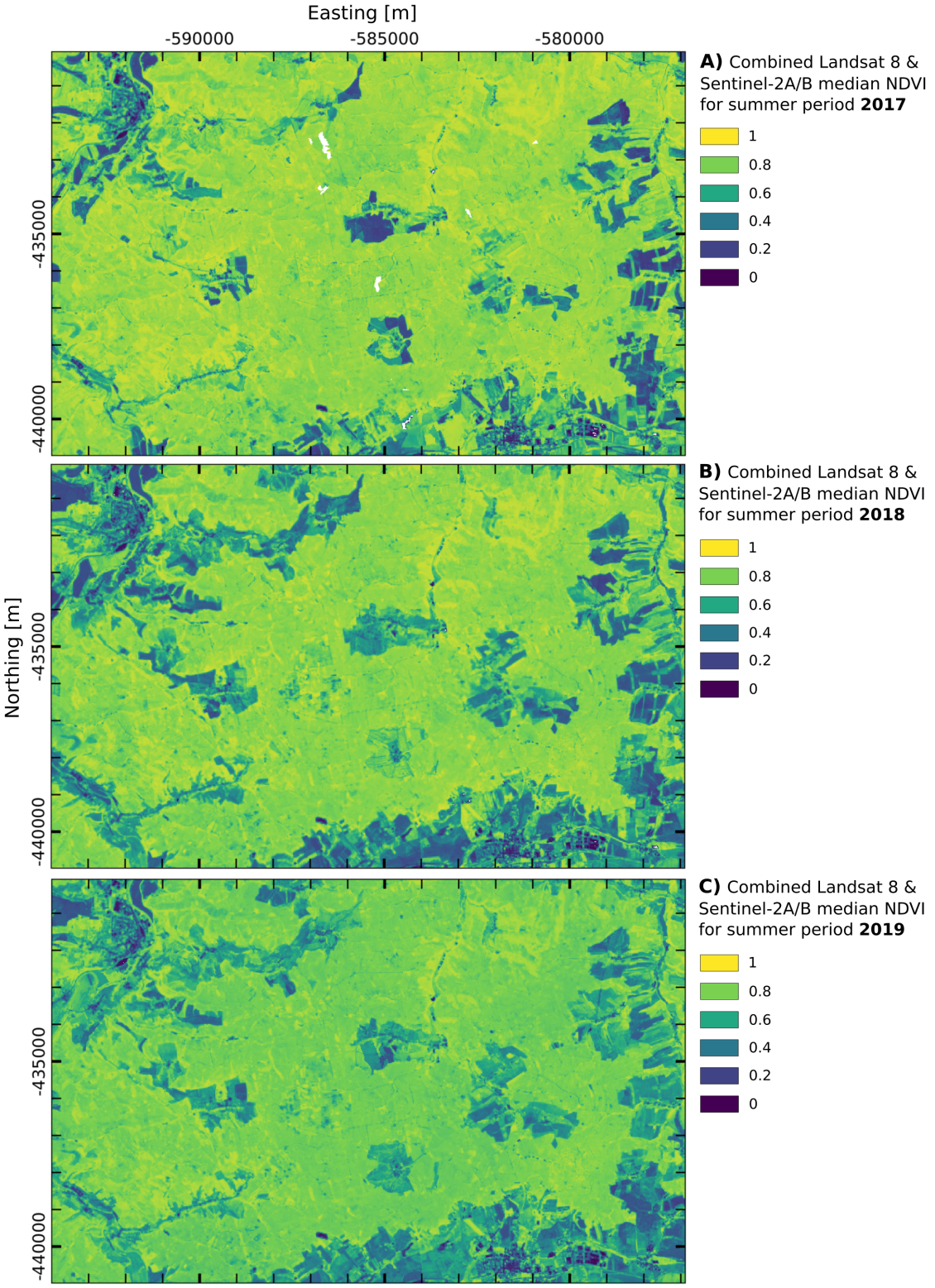

Figure B-5: Individual plots of median NDVI (Normalized Difference Vegetation Index) derived from the optical datasets for the summer periods (June, July, August) of 2017 (A), 2018 (B) and 2019 (C).

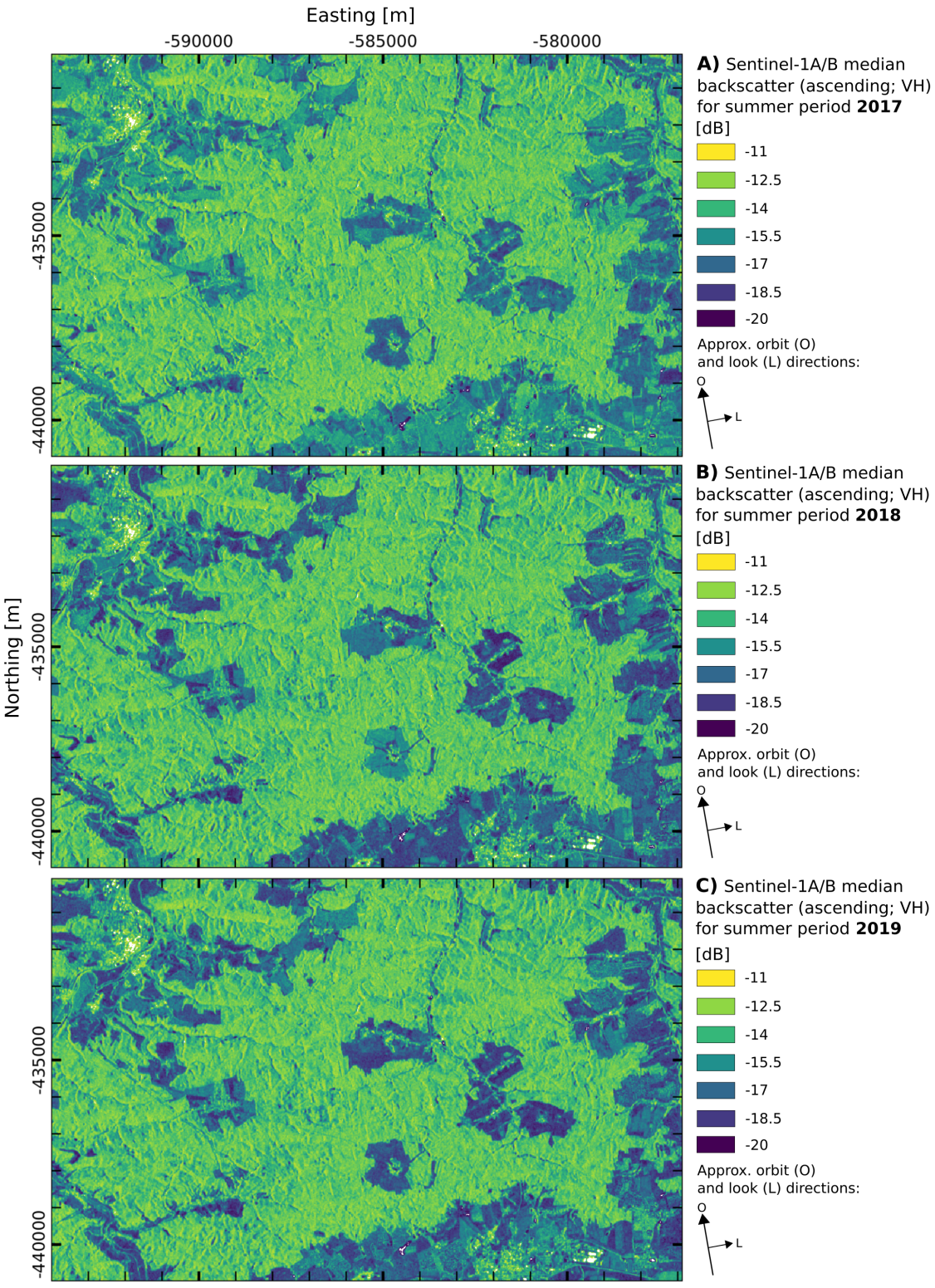

Figure B-6: Individual plots of median SAR backscatter (ascending orbit; VH polarization) for the summer periods (June, July, August) of 2017 (A), 2018 (B) and 2019 (C).

References

Abernathey, R., Augspurger, T., Banihirwe, A., Blackmon-Luca, C. C., Crone, T. J., Gentemann, C. L., Hamman, J. J., Henderson, N., Lepore, C., Mccaie, T. A., Robinson, N. H., & Signell, R. P. (2020). Cloud-Native Repositories for Big Scientific Data. *Computing in Science & Engineering*, *23*, 26–35. https://doi.org/10.22541/au.160443768.88917719/v1

Alberti, K. (2018, August 16). *Guide to GeoTIFF compression and optimization with GDAL*. https://kokoalberti.com/articles/geotiff-compression-optimization-guide/

Anaconda, Inc. (2017). *Conda: Package, dependency and environment management for any language—python, r, ruby, lua, scala, java, JavaScript, c/ c++, FORTRAN, and more.* Anaconda, Inc. https://conda.io

Anaconda, Inc., & Contributors. (n.d.-a). *Dask Documentation: Chunks*. Retrieved May 17, 2021, from https://docs.dask.org/en/latest/array-chunks.html

Anaconda, Inc., & Contributors. (n.d.-b). *Dask Documentation: Diagnosing performance*. Retrieved June 12, 2021, from https://distributed.dask.org/en/latest/diagnosing-performance.html

Anderson, K., Ryan, B., Sonntag, W., Kavvada, A., & Friedl, L. (2017). Earth observation in service of the 2030 Agenda for Sustainable Development. *Geo-Spatial Information Science*, *20*(2), 77–96. https://doi.org/10.1080/10095020.2017.1333230

Appel, M., & Pebesma, E. (2019). On-Demand Processing of Data Cubes from Satellite Image Collections with the gdalcubes Library. *Data*, *4*(3), 92. https://doi.org/10.3390/data4030092

Ariav, G. (1986). A temporally oriented data model. *ACM Transactions on Database Systems*, *11*(4), 499–527. https://doi.org/10.1145/7239.7350

Aschbacher, J. (2017). ESA's Earth Observation Strategy and Copernicus. In *Satellite Earth Observations and Their Impact on Society and Policy* (pp. 81–86). Springer Singapore. https://doi.org/10.1007/978-981-10-3713-9_5

Asmaryan, S., Muradyan, V., Tepanosyan, G., Hovsepyan, A., Saghatelyan, A., Astsatryan, H., Grigoryan, H., Abrahamyan, R., Guigoz, Y., & Giuliani, G. (2019). Paving the Way towards an Armenian Data Cube. *Data*, *4*(3), 117. https://doi.org/10.3390/data4030117

Barker, N. (2020, June 23). *Open SAR data and scalable analytics: TileDB joins forces with capella space on building SAR developer communities*. TileDB, Inc. https://tiledb.com/blog/open-sar-data-and-scalable-analytics-2020-06-23

Bauer-Marschallinger, B., Sabel, D., & Wagner, W. (2014). Optimisation of global grids for high-resolution remote sensing data. *Computers & Geosciences*, *72*, 84–93. https://doi.org/10.1016/j.cageo.2014.07.005

Baumann, P. (2017). *The Datacube Manifesto*. https://www.earthserver.eu/tech/datacube-manifesto/The-Datacube-Manifesto.pdf

Baumann, P., Dehmel, A., Furtado, P., Ritsch, R., & Widmann, N. (1998). The multidimensional database system RasDaMan. *Proceedings of the 1998 ACM SIGMOD international conference on Management of data - SIGMOD 98*. https://doi.org/10.1145/276304.276386

Baumann, P., Mazzetti, P., Ungar, J., Barbera, R., Barboni, D., Beccati, A., Bigagli, L., Boldrini, E., Bruno, R., Calanducci, A., Campalani, P., Clements, O., Dumitru, A., Grant, M., Herzig, P., Kakaletris, G., Laxton, J., Koltsida, P., Lipskoch, K., … Wagner, S. (2015). Big Data Analytics for Earth Sciences: the EarthServer approach. *International Journal of Digital Earth*, *9*(1), 3–29. https://doi.org/10.1080/17538947.2014.1003106

Baumann, P., Misev, D., Merticariu, V., & Huu, B. P. (2019). Datacubes: Towards Space/Time Analysis-Ready Data. In *Service-Oriented Mapping: Changing Paradigm in Map Production and Geoinformation Management* (pp. 269–299). Springer International Publishing. https://doi.org/10.1007/978-3-319-72434-8_14

Bloss, A., Hudak, P., & Young, J. (1988). Code optimizations for lazy evaluation. *Lisp and Symbolic Computation*, *1*(2), 147–164. https://doi.org/10.1007/bf01806169

Boettiger, C. (2015). An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review*, *49*(1), 71–79. https://doi.org/10.1145/2723872.2723882

Boulton, G. (2018). The challenges of a Big Data Earth. *Big Earth Data*, *2*(1), 1–7. https://doi.org/10.1080/20964471.2017.1397411

Bouvet, A., Mermoz, S., Ballère, M., Koleck, T., & Toan, T. L. (2018). Use of the SAR Shadowing Effect for Deforestation Detection with Sentinel-1 Time Series. *Remote Sensing*, *10*(8), 1250. https://doi.org/10.3390/rs10081250

Bravo, G., Castro, H., Moreno, A., Ariza-Porras, C., Galindo, G., Cabrera, E., Valbuena, S., & Lozano-Rivera, P. (2017). Architecture for a Colombian Data Cube Using Satellite Imagery for Environmental Applications. In A. Solano & H. Ordoñez (Eds.), *Advances in Computing* (pp. 227–241). Springer International Publishing.

Brazil Data Cube Contributors. (n.d.). *Brazil Data Cube: Github Organization*. National Institute for Space Research (INPE). Retrieved June 10, 2021, from https://github.com/brazil-data-cube

Bunting, P., & Contributors. (n.d.). *Atmospheric and Radiometric Correction of Satellite Imagery (ARCSI)* [Computer software]. Retrieved May 10, 2021, from https://github.com/remotesensinginfo/arcsi

CEOS. (n.d.-a). *CEOS Analysis Ready Data: Analysis ready data resources*. Committee on Earth Observation Satellites. Retrieved June 16, 2021, from https://ceos.org/ard/resources.html

CEOS. (n.d.-b). *CEOS Analysis Ready Data: Product family specifications*. Committee on Earth Observation Satellites. Retrieved June 10, 2021, from https://ceos.org/ard/index.html#slide3

Claverie, M., Ju, J., Masek, J. G., Dungan, J. L., Vermote, E. F., Roger, J.-C., Skakun, S. V., & Justice, C. (2018). The Harmonized Landsat and Sentinel-2 surface reflectance data set. *Remote Sensing of Environment*, *219*, 145–161. https://doi.org/10.1016/j.rse.2018.09.002

Coetzee, S., Ivánová, I., Mitasova, H., & Brovelli, M. A. (2020). Open Geospatial Software and Data: A Review of the Current State and A Perspective into the Future. *ISPRS International Journal of Geo-Information*, *9*(2, 2), 90. https://doi.org/10.3390/ijgi9020090

configparser Contributors. (n.d.). *Configparser: Configuration file parser* [Computer software]. Python Software Foundation. Retrieved May 12, 2021, from https://docs.python.org/3/library/configparser.html

Corbly, J. E. (2014). The free software alternative: Freeware, open source software, and libraries. *Information Technology and Libraries*, *33*(3), 65. https://doi.org/10.6017/ital.v33i3.5105

Dask-ML Contributors. (n.d.). *Dask-ML: Scalable machine learning with dask* [Computer software]. Retrieved May 28, 2021, from https://github.com/dask/dask-ml

de Araujo Barbosa, C. C., Atkinson, P. M., & Dearing, J. A. (2015). Remote sensing of ecosystem services: A systematic review. *Ecological Indicators*, *52*, 430–443. https://doi.org/10.1016/j.ecolind.2015.01.007

Destatis. (2021, April 21). *Amount of timber logged at new record high in 2020 due to forest damage: Timber infested by insects accounted for more than half of the total amount of timber logged*. https://www.destatis.de/EN/Press/2021/04/PE21_192_413.html

Dhar, T. (2021, April 22). *ODC Enhancement Proposal 004: Use alternative index backends*. https://github.com/opendatacube/datacube-core/wiki/ODC-EP-004---Use-alternative-index-backends

Dhu, T., Giuliani, G., Juárez, J., Kavvada, A., Killough, B., Merodio, P., Minchin, S., & Ramage, S. (2019). National Open Data Cubes and Their Contribution to Country-Level Development Policies and Practices. *Data*, *4*(4), 144. https://doi.org/10.3390/data4040144

Diaz, L., & Remke, A. (2012). Future SDI – Impulses from Geoinformatics Research and IT Trends. *International Journal of Spatial Data Infrastructures Research*, *7*, 378–410. https://doi.org/10.2902/1725-0463.2012.07.art18

Digital Earth Africa. (n.d.). *Digital Earth Africa*. Retrieved June 10, 2021, from https://www.digitalearthafrica.org/

Digital Earth Australia. (n.d.). *DEA Coastlines Map*. Geoscience Australia. Retrieved June 8, 2021, from https://maps.dea.ga.gov.au/#share=s-DEACoastlines&playStory=1

Docker. (n.d.). *What is a Container?* Retrieved June 4, 2021, from https://www.docker.com/resources/what-container

Dubois, C., Mueller, M. M., Pathe, C., Jagdhuber, T., Cremer, F., Thiel, C., & Schmullius, C. (2020). Characterization of Land Cover Seasonality in Sentinel-1 Time Series Data. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *V-3-2020*, 97–104. https://doi.org/10.5194/isprs-annals-v-3-2020-97-2020

Dwyer, J., Roy, D., Sauer, B., Jenkerson, C., Zhang, H., & Lymburner, L. (2018). Analysis Ready Data: Enabling Analysis of the Landsat Archive. *Remote Sensing*, *10*(9). https://doi.org/10.3390/rs10091363

Eckman, R. S., & Stackhouse, P. W. (2012). CEOS contributions to informing energy management and policy decision making using space-based Earth observations. *Applied Energy*, *90*(1), 206–210. https://doi.org/10.1016/j.apenergy.2011.03.001

ESA. (n.d.-a). *Sentinel-2 Data Products*. European Space Agency. Retrieved June 6, 2021, from https://sentinel.esa.int/web/sentinel/missions/sentinel-2/data-products

ESA. (n.d.-b). *Sentinel-2 Revisit and Coverage*. European Space Agency. Retrieved June 6, 2021, from https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/revisit-coverage

ESA. (n.d.-c). *SNAP: Sentinel Application Platform*. European Space Agency. Retrieved June 10, 2021, from http://step.esa.int/main/toolboxes/snap/

ESA. (2021, March 22). *Forthcoming deployment of the Copernicus Sentinel-2 products geometric refinement*. European Space Agency. https://sentinel.esa.int/web/sentinel/-/forthcoming-deployment-of-the-copernicus-sentinel-2-products-geometric-refinement/1.1

ESDL. (n.d.). *Earth System Data Lab*. Brockmann Consult GmbH. Retrieved June 10, 2021, from https://www.earthsystemdatalab.net/

Eurac Research, & Dares Technology. (n.d.). *SAR2CUBE*. Retrieved June 2, 2021, from https://eo4society.esa.int/projects/sar2cube/

Euro Data Cube Consortium. (n.d.). *Euro Data Cube*. Retrieved June 10, 2021, from https://www.eurodatacube.com/

European Commission. (2013, July 12). *Commission Delegated Regulation (EU) No 1159/2013 of 12 July 2013 supplementing Regulation (EU) No 911/2010 of the European Parliament and of the Council on the European Earth monitoring programme (GMES) by establishing registration and licensing conditions for GMES users and defining criteria for restricting access to GMES dedicated data and GMES service information*. http://data.europa.eu/eli/reg_del/2013/1159/oj

Eynard-Bontemps, G., Abernathey, R., Hamman, J., Ponte, A., & Rath, W. (2019). The Pangeo Big Data Ecosystem and its use at CNES. In P. Soille, S. Loekken, & S. Albani (Eds.), *Proceedings of the 2019 conference on Big Data from Space (BiDS'19)* (pp. 49–52). Publications Office of the European Union. https://doi.org/10.2760/848593

Farquharson, G., Woods, W., Stringham, C., Sankarambadi, N., & Riggi, L. (2018). The Capella Synthetic Aperture Radar Constellation. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 1873–1876. https://doi.org/10.1109/igarss.2018.8518683

Fernandez-Carrillo, A., Patočka, Z., Dobrovolný, L., Franco-Nieto, A., & Revilla-Romero, B. (2020). Monitoring Bark Beetle Forest Damage in Central Europe. A Remote Sensing Approach Validated with Field Data. *Remote Sensing*, *12*(21), 3634. https://doi.org/10.3390/rs12213634

Ferreira, K. R., Queiroz, G. R., Camara, G., Souza, R. C. M., Vinhas, L., Marujo, R. F. B., Simoes, R. E. O., Noronha, C. A. F., Costa, R. W., Arcanjo, J. S., Gomes, V. C. F., & Zaglia, M. C. (2020). Using Remote Sensing Images and Cloud Services on Aws to Improve Land Use and Cover Monitoring. *2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS)*, 558–562. https://doi.org/10.1109/lagirs48042.2020.9165649

Ferreira, K. R., Queiroz, G. R., Vinhas, L., Marujo, R. F. B., Simoes, R. E. O., Picoli, M. C. A., Camara, G., Cartaxo, R., Gomes, V. C. F., Santos, L. A., Sanchez, A. H., Arcanjo, J. S., Fronza, J. G., Noronha, C. A., Costa, R. W., Zaglia, M. C., Zioti, F., Korting, T. S., Soares, A. R., … Fonseca, L. M. G. (2020). Earth Observation Data Cubes for Brazil: Requirements, Methodology and Products. *Remote Sensing*, *12*(24), 4033. https://doi.org/10.3390/rs12244033

Forzieri, G., Cescatti, A., Silva, F. B. e, & Feyen, L. (2017). Increasing risk over time of weather-related hazards to the European population: a data-driven prognostic study. *The Lancet Planetary Health*, *1*(5), e200–e208. https://doi.org/10.1016/s2542-5196(17)30082-7

Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008). Cloud Computing and Grid Computing 360-Degree Compared. *2008 Grid Computing Environments Workshop*, 1–10. https://doi.org/10.1109/gce.2008.4738445

Frank, D., Reichstein, M., Bahn, M., Thonicke, K., Frank, D., Mahecha, M. D., Smith, P., Velde, M., Vicca, S., Babst, F., Beer, C., Buchmann, N., Canadell, J. G., Ciais, P., Cramer, W., Ibrom, A., Miglietta, F., Poulter, B., Rammig, A., … Zscheischler, J. (2015). Effects of climate extremes on the terrestrial carbon cycle: concepts, processes and potential future impacts. *Global Change Biology*, *21*(8), 2861–2880. https://doi.org/10.1111/gcb.12916

Frantz, D. (2019). FORCE—Landsat + Sentinel-2 Analysis Ready Data and Beyond. *Remote Sensing*, *11*(9), 1124. https://doi.org/10.3390/rs11091124

Frantz, D., & Contributors. (n.d.-a). *FORCE Documentation: Level 1 Archiving Suite*. Retrieved May 4, 2021, from https://force-eo.readthedocs.io/en/latest/components/lower-level/level1/index.html

Frantz, D., & Contributors. (n.d.-b). *FORCE Documentation: Level 2 Processing System*. Retrieved May 4, 2021, from https://force-eo.readthedocs.io/en/latest/components/lower-level/level2/index.html

Frantz, D., & Contributors. (n.d.-c). *FORCE Documentation: Tutorial: Level 2 ARD*. Retrieved May 4, 2021, from https://force-eo.readthedocs.io/en/latest/howto/l2-ard.html#tut-ard

Frantz, D., & Contributors. (n.d.-d). *FORCE: Framework for Operational Radiometric Correction for Environmental monitoring* [Computer software]. Retrieved May 8, 2021, from https://github.com/davidfrantz/force

Frantz, D., Haß, E., Uhl, A., Stoffels, J., & Hill, J. (2018). Improvement of the Fmask algorithm for Sentinel-2 images: Separating clouds from bright surfaces based on parallax effects. *Remote Sensing of Environment*, *215*, 471–481. https://doi.org/10.1016/j.rse.2018.04.046

Frantz, D., Roder, A., Stellmes, M., & Hill, J. (2016). An Operational Radiometric Landsat Preprocessing Framework for Large-Area Time Series Applications. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(7), 3928–3943. https://doi.org/10.1109/tgrs.2016.2530856

Frantz, D., Roder, A., Udelhoven, T., & Schmidt, M. (2015). Enhancing the Detectability of Clouds and Their Shadows in Multitemporal Dryland Landsat Imagery: Extending Fmask. *IEEE Geoscience and Remote Sensing Letters*, *12*(6), 1242–1246. https://doi.org/10.1109/lgrs.2015.2390673

Frantz, D., Stellmes, M., & Hostert, P. (2019). A Global MODIS Water Vapor Database for the Operational Atmospheric Correction of Historic and Recent Landsat Imagery. *Remote Sensing*, *11*(3), 257. https://doi.org/10.3390/rs11030257

Frantz, D., Stellmes, M., Roder, A., Udelhoven, T., Mader, S., & Hill, J. (2016). Improving the Spatial Resolution of Land Surface Phenology by Fusing Medium- and Coarse-Resolution Inputs. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(7), 4153–4164. https://doi.org/10.1109/tgrs.2016.2537929

Freistaat Thüringen. (2018, December 29). *Thüringer Gesetz zum Klimaschutz und zur Anpassung an die Folgen des Klimawandels*. https://landesrecht.thueringen.de/perma?a=KlimaSchG_TH

Friedl, M., Olofsson, P., Woodcock, C., & others. (n.d.). *NASA MEaSUREs 2018-2023: A Data Record of 21st Century Global Land Cover/Use/Change*. Boston University. Retrieved May 15, 2021, from http://sites.bu.edu/measures/

Frischbier, N., Profft, I., & Hagemann, U. (2013). Potential Impacts of Climate Change on Forest Habitats in the Biosphere Reserve Vessertal-Thuringian Forest in Germany. In *Advances in Global Change Research* (pp. 243–257). Springer Netherlands. https://doi.org/10.1007/978-94-007-7960-0_16

GAMMA Remote Sensing. (n.d.). *GAMMA Software* [Computer software]. GAMMA Remote Sensing Research and Consulting AG. Retrieved May 1, 2021, from https://www.gamma-rs.ch/

Garofoli, A., Paradiso, V., Montazeri, H., Jermann, P. M., Roma, G., Tornillo, L., Terracciano, L. M., Piscuoglio, S., & Ng, C. K. Y. (2019). PipeIT: A Singularity Container for Molecular Diagnostic Somatic Variant Calling on the Ion Torrent Next-Generation Sequencing Platform. *The Journal of Molecular Diagnostics*, *21*(5), 884–894. https://doi.org/10.1016/j.jmoldx.2019.05.001

Gascon, F., Bouzinac, C., Thépaut, O., Jung, M., Francesconi, B., Louis, J., Lonjou, V., Lafrance, B., Massera, S., Gaudel-Vacaresse, A., Languille, F., Alhammoud, B., Viallefont, F., Pflug, B., Bieniarz, J., Clerc, S., Pessiot, L., Trémas, T., Cadau, E., … Fernandez, V. (2017). Copernicus Sentinel-2A Calibration and Products Validation Status. *Remote Sensing*, *9*(6), 584. https://doi.org/10.3390/rs9060584

Gentemann, C. L., Holdgraf, C., Abernathey, R., Crichton, D., Colliander, J., Kearns, E. J., Panda, Y., & Signell, R. P. (2021). Science Storms the Cloud. *AGU Advances*, *2*(2). https://doi.org/10.1029/2020av000354

GeoBasis-DE / BKG. (2021). *Verwaltungsgebiete 1:2 500 000, Stand 01.01. (VG2500)*. https://daten.gdz.bkg.bund.de/produkte/vg/vg2500/aktuell/vg2500_01-01.utm32s.shape.zip

Geoportal-Th. (2021a, March 30). *ATKIS Basis-DLM*. Freistaat Thüringen - Landesamt für Bodenmanagement und Geoinformation. https://www.geoportal-th.de/de-de/Downloadbereiche/Download-Offene-Geodaten-Thüringen/Download-ATKIS-Basis-DLM

Geoportal-Th. (2021b). *Höhendaten (DGM / DOM / LAZ)*. Freistaat Thüringen - Landesamt für Bodenmanagement und Geoinformation. https://www.geoportal-th.de/de-de/Downloadbereiche/Download-Offene-Geodaten-Thüringen/Download-Höhendaten

Giuliani, G., Camara, G., Killough, B., & Minchin, S. (2019). Earth Observation Open Science: Enhancing Reproducible Science Using Data Cubes. *Data*, *4*(4), 147. https://doi.org/10.3390/data4040147

Giuliani, G., Chatenoux, B., Benvenuti, A., Lacroix, P. M. A., Santoro, M., & Mazzetti, P. (2020). Monitoring land degradation at national level using satellite Earth Observation time-series data to support SDG15 – exploring the potential of data cube. *Big Earth Data*, *4*, 3–22. https://doi.org/10.1080/20964471.2020.1711633

Giuliani, G., Chatenoux, B., Bono, A. D., Rodila, D., Richard, J.-P., Allenbach, K., Dao, H., & Peduzzi, P. (2017). Building an Earth Observations Data Cube: lessons learned from the Swiss Data Cube (SDC) on generating Analysis Ready Data (ARD). *Big Earth Data*, *1*(1-2), 100–117. https://doi.org/10.1080/20964471.2017.1398903

Giuliani, G., Chatenoux, B., Honeck, E., & Richard, J.-P. (2018). Towards Sentinel-2 Analysis Ready Data: a Swiss Data Cube Perspective. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 8659–8662. https://doi.org/10.1109/igarss.2018.8517954

Giuliani, G., Dao, H., Bono, A. D., Chatenoux, B., Allenbach, K., Laborie, P. D., Rodila, D., Alexandris, N., & Peduzzi, P. (2017). Live Monitoring of Earth Surface (LiMES): A framework for monitoring environmental changes from Earth Observations. *Remote Sensing of Environment*, *202*, 222–233. https://doi.org/10.1016/j.rse.2017.05.040

Giuliani, G., Egger, E., Italiano, J., Poussin, C., Richard, J.-P., & Chatenoux, B. (2020). Essential Variables for Environmental Monitoring: What Are the Possible Contributions of Earth Observation Data Cubes? *Data*, *5*, 100. https://doi.org/10.3390/data5040100

Giuliani, G., Masó, J., Mazzetti, P., Nativi, S., & Zabala, A. (2019). Paving the Way to Increased Interoperability of Earth Observations Data Cubes. *Data*, *4*(3), 113. https://doi.org/10.3390/data4030113

Gomes, V. C. F., Queiroz, G. R. de, & Ferreira, K. R. (2020). An Overview of Platforms for Big Earth Observation Data Management and Analysis. *Remote Sensing*, *12*, 1253. https://doi.org/10.3390/rs12081253

Gore, A. (1998). The Digital Earth. *Australian Surveyor*, *43*(2), 89–91. https://doi.org/10.1080/00050348.1998.10558728

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, *202*, 18–27. https://doi.org/10.1016/j.rse.2017.06.031

Gowda, S. (n.d.). *Open Earth Alliance: 2020-2022 GEO Work Programme*. Retrieved May 31, 2021, from https://www.earthobservations.org/documents/gwp20_22/OEA.pdf

Gowda, S., & Killough, B. (2020). Open Data Cube (ODC) Visualization: Bridging the Gap between Data, Decisions, and Development Goals. *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, 3395–3398. https://doi.org/10.1109/igarss39084.2020.9324669

Guo, H., Liu, Z., Jiang, H., Wang, C., Liu, J., & Liang, D. (2016). Big Earth Data: a new challenge and opportunity for Digital Earth's development. *International Journal of Digital Earth*, *10*(1), 1–12. https://doi.org/10.1080/17538947.2016.1264490

Gutzmann, B., Motl, A., Lassahn, D., Smith, T. J., Kamenshchikov, I., Schrammel, M., & Contributors. (n.d.). *Wetterdienst: Open weather data for humans* [Computer software]. Retrieved May 28, 2021, from https://github.com/earthobservations/wetterdienst

Hamman, J. (2020, March 3). *Introducing Intake-stac*. https://medium.com/pangeo/introducing-intake-stac-2c988d8e1d30

Hansen, M. C., Potapov, P. V., Moore, R., Hancher, M., Turubanova, S. A., Tyukavina, A., Thau, D., Stehman, S. V., Goetz, S. J., Loveland, T. R., Kommareddy, A., Egorov, A., Chini, L., Justice, C. O., & Townshend, J. R. G. (2013). High-Resolution Global Maps of 21st-Century Forest Cover Change. *Science*, *342*(6160), 850–853. https://doi.org/10.1126/science.1244693

Hernández-López, D., Piedelobo, L., Moreno, M. A., Chakhar, A., Ortega-Terol, D., & González-Aguilera, D. (2021). Design of a Local Nested Grid for the Optimal Combined Use of Landsat 8 and Sentinel 2 Data. *Remote Sensing*, *13*(8), 1546. https://doi.org/10.3390/rs13081546

Hilbert, M., & Lopez, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, *332*(6025), 60–65. https://doi.org/10.1126/science.1200970

Holden, C. (n.d.). *GLANCE Grids: Global LANd Cover mapping and Estimation (GLANCE) Grids*. Boston University. Retrieved June 4, 2021, from https://measures-glance.github.io/glance-grids/

Hollaus, M., & Vreugdenhil, M. (2019). Radar Satellite Imagery for Detecting Bark Beetle Outbreaks in Forests. *Current Forestry Reports*, *5*(4), 240–250. https://doi.org/10.1007/s40725-019-00098-z

Hollmann, R., Merchant, C. J., Saunders, R., Downy, C., Buchwitz, M., Cazenave, A., Chuvieco, E., Defourny, P., Leeuw, G. de, Forsberg, R., Holzer-Popp, T., Paul, F., Sandven, S., Sathyendranath, S., Roozendael, M. van, & Wagner, W. (2013). The ESA Climate Change Initiative: Satellite Data Records for Essential Climate Variables. *Bulletin of the American Meteorological Society*, *94*(10), 1541–1552. https://doi.org/10.1175/bams-d-11-00254.1

Hoyer, S., & Hamman, J. J. (2017). xarray: N-D labeled Arrays and Datasets in Python. *Journal of Open Research Software*, *5*. https://doi.org/10.5334/jors.148

Inglada, J., & Christophe, E. (2009). The Orfeo Toolbox remote sensing image processing software. *2009 IEEE International Geoscience and Remote Sensing Symposium*, pp. IV-733-IV-736. https://doi.org/10.1109/igarss.2009.5417481

Joseph, G., & Contributors. (n.d.). *StackSTAC: Turn a STAC catalog into a dask-based xarray* [Computer software]. Retrieved May 31, 2021, from https://github.com/gjoseph92/stackstac

Kampe, T. U., & Good, W. S. (2017). Pathway to future sustainable land imaging: the compact hyperspectral prism spectrometer. In J. J. Butler, X. (Jack) Xiong, & X. Gu (Eds.), *Earth Observing Systems XXII* (Vol. 10402, pp. 74–84). SPIE. https://doi.org/10.1117/12.2270932

Kellndorfer, J. (2021). *The new era of SAR time series: Tackling big EO data analysis and visualization with Pangeo tools*. https://doi.org/10.5281/ZENODO.4756696

Kellogg, K., Hoffman, P., Standley, S., Shaffer, S., Rosen, P., Edelstein, W., Dunn, C., Baker, C., Barela, P., Shen, Y., Guerrero, A. M., Xaypraseuth, P., Sagi, V. R., Sreekantha, C. V., Harinath, N., Kumar, R., Bhan, R., & Sarma, C. V. H. S. (2020). NASA-ISRO Synthetic Aperture Radar (NISAR) Mission. *2020 IEEE Aerospace Conference*, 1–21. https://doi.org/10.1109/aero47225.2020.9172638

Killough, B. (2018). Overview of the Open Data Cube Initiative. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 8629–8632. https://doi.org/10.1109/igarss.2018.8517694

Killough, B. (2019). *The Impact of Analysis Ready Data in the Africa Regional Data Cube*. 5646–5649. https://doi.org/10.1109/igarss.2019.8898321

Kopp, S., Becker, P., Doshi, A., Wright, D. J., Zhang, K., & Xu, H. (2019). Achieving the Full Vision of Earth Observation Data Cubes. *Data*, *4*(3), 94. https://doi.org/10.3390/data4030094

Kouzoubov, K., Ayers, D., & Leith, A. (2019, June 12). *ODC Enhancement Proposal 003: Replace the ODC Index and Internal Database API.* Open Data Cube. https://github.com/opendatacube/datacube-core/wiki/ODC-EP-003---Replace-the-ODC-Index-and-Internal-Database-API

Krause, C., Dunn, B., & Bishop-Taylor, R. (2021). *Digital Earth Australia notebooks and tools repository.* Commonwealth of Australia (Geoscience Australia). https://doi.org/10.26186/145234

Kron, W., Löw, P., & Kundzewicz, Z. W. (2019). Changes in risk of extreme weather events in Europe. *Environmental Science & Policy*, *100*, 74–83. https://doi.org/10.1016/j.envsci.2019.06.007

Kurtzer, G. M., Sochat, V., & Bauer, M. W. (2017). Singularity: Scientific containers for mobility of compute. *PLOS ONE*, *12*(5), e0177459. https://doi.org/10.1371/journal.pone.0177459

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: a LLVM-based Python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC - LLVM '15*, 1–6. https://doi.org/10.1145/2833157.2833162

Laney, D. (2001). 3D data management: Controlling data volume, velocity and variety. *META Group Research Note*, *6*(70), 1.

Leith, A. (2018, August 9). *What is the Open Data Cube?* https://medium.com/opendatacube/what-is-open-data-cube-805af60820d7

Lewis, A., Lacey, J., Mecklenburg, S., Ross, J., Siqueira, A., Killough, B., Szantoi, Z., Tadono, T., Rosenavist, A., Goryl, P., Miranda, N., & Hosford, S. (2018). CEOS Analysis Ready Data for Land (CARD4L) Overview. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 7407–7410. https://doi.org/10.1109/igarss.2018.8519255

Lewis, A., Lymburner, L., Purss, M. B. J., Brooke, B. P., Evans, B. J. K., Ip, A., Dekker, A. G., Irons, J. R., Minchin, S., Mueller, N., Oliver, S., Roberts, D., Ryan, B., Thankappan, M., Woodcock, R., & Wyborn, L. (2016). Rapid, high-resolution detection of environmental change over continental scales from satellite data – the Earth Observation Data Cube. *International Journal of Digital Earth*, *9*, 106–111. https://doi.org/10.1080/17538947.2015.1111952

Lewis, A., Oliver, S., Lymburner, L., Evans, B., Wyborn, L., Mueller, N., Raevksi, G., Hooke, J., Woodcock, R., Sixsmith, J., Wu, W., Tan, P., Li, F., Killough, B., Minchin, S., Roberts, D., Ayers, D., Bala, B., Dwyer, J., … Wang, L.-W. (2017). The Australian Geoscience Data Cube — Foundations and lessons learned. *Remote Sensing of Environment*, *202*, 276–292. https://doi.org/10.1016/j.rse.2017.03.015

Lonjou, V., Desjardins, C., Hagolle, O., Petrucci, B., Tremas, T., Dejus, M., Makarau, A., & Auer, S. (2016). MACCS-ATCOR joint algorithm (MAJA). In A. Comerón, E. I. Kassianov, & K. Schäfer (Eds.), *Remote Sensing of Clouds and the Atmosphere XXI* (Vol. 1000107, pp. 25–37). SPIE. https://doi.org/10.1117/12.2240935

Mahecha, M. D., Gans, F., Brandt, G., Christiansen, R., Cornell, S. E., Fomferra, N., Kraemer, G., Peters, J., Bodesheim, P., Camps-Valls, G., Donges, J. F., Dorigo, W., Estupiñán-Suárez, L. M., Gutierrez-Velez, V. H., Gutwin, M., Jung, M., Londoño, M. C., Miralles, D. G., Papastefanou, P., & Reichstein, M. (2020). Earth system data cubes unravel global multivariate dynamics. *Earth System Dynamics Discussions*, *11*, 201–234. https://doi.org/10.5194/esd-11-201-2020

Main-Knorn, M., Pflug, B., Louis, J., Debaecker, V., Müller-Wilm, U., & Gascon, F. (2017). Sen2Cor for Sentinel-2. In L. Bruzzone, F. Bovolo, & J. A. Benediktsson (Eds.), *Image and Signal Processing for Remote Sensing XXIII* (Vol. 1042704). SPIE. https://doi.org/10.1117/12.2278218

MapBox, Inc., & Contributors. (n.d.). *Rasterio: geospatial raster I/O for Python programmers* [Computer software]. Mapbox. Retrieved May 12, 2021, from https://github.com/mapbox/rasterio

Masek, J. G., Wulder, M. A., Markham, B., McCorkel, J., Crawford, C. J., Storey, J., & Jenstrom, D. T. (2020). Landsat 9: Empowering open science and applications through continuity. *Remote Sensing of Environment*, *248*, 111968. https://doi.org/10.1016/j.rse.2020.111968

Maso, J., Zabala, A., Serral, I., & Pons, X. (2019). A Portal Offering Standard Visualization and Analysis on top of an Open Data Cube for Sub-National Regions: The Catalan Data Cube Example. *Data*, *4*(3), 96. https://doi.org/10.3390/data4030096

Misev, D., Baumann, P., Bellos, D., & Wiehle, S. (2019). BigDataCube: A Scalable, Federated Service Platform for Copernicus. *2019 IEEE International Conference on Big Data (Big Data)*, 4103–4112. https://doi.org/10.1109/bigdata47090.2019.9006222

Mölder, F., Jablonski, K. P., Letcher, B., Hall, M. B., Tomkins-Tinch, C. H., Sochat, V., Forster, J., Lee, S., Twardziok, S. O., Kanitz, A., Wilm, A., Holtgrewe, M., Rahmann, S., Nahnsen, S., & Köster, J. (2021). Sustainable data analysis with Snakemake. *F1000Research*, *10*, 33. https://doi.org/10.12688/f1000research.29032.2

Nativi, S., Mazzetti, P., & Craglia, M. (2017). A view-based model of data-cube to support big earth data systems interoperability. *Big Earth Data*, *1*(1-2), 75–99. https://doi.org/10.1080/20964471.2017.1404232

Nemirovsky, M., & Tullsen, D. M. (2013). Multithreading Architecture. *Synthesis Lectures on Computer Architecture*, *8*(1), 1–109. https://doi.org/10.2200/s00458ed1v01y201212cac021

ODC Contributors. (n.d.-a). *Open Data Cube Core* [Computer software]. Retrieved May 10, 2021, from https://github.com/opendatacube/datacube-core

ODC Contributors. (n.d.-b). *Open Data Cube Documentation: Data Cube Ecosystem*. Open Data Cube. Retrieved June 10, 2021, from https://datacube-core.readthedocs.io/en/latest/user/ecosystem.html

ODC Contributors. (n.d.-c). *Open Data Cube Documentation: Dataset Documents - EO3 Format*. Open Data Cube. Retrieved June 6, 2021, from https://datacube-core.readthedocs.io/en/latest/ops/dataset_documents.html#eo3-format

ODC Contributors. (n.d.-d). *Open Data Cube Documentation: High Level Architecture*. Open Data Cube. Retrieved April 26, 2021, from https://datacube-core.readthedocs.io/en/latest/architecture/high_level.html

ODC Contributors. (n.d.-e). *Open Data Cube Documentation: Product Definition*. Open Data Cube. Retrieved June 6, 2021, from https://datacube-core.readthedocs.io/en/latest/ops/product.html

OSGeo. (n.d.). *OSGeo Community Projects: Open Data Cube*. Retrieved May 31, 2021, from https://www.osgeo.org/projects/open-data-cube/

Pangeo Contributors. (n.d.-a). *Pangeo: A community platform for Big Data geoscience*. Retrieved May 31, 2021, from https://pangeo.io/index.html

Pangeo Contributors. (n.d.-b). *Pangeo Documentation: Packages - Why Xarray and Dask?* Retrieved May 8, 2021, from https://pangeo.io/packages.html#why-xarray-and-dask

Papadopoulos, S., Datta, K., Madden, S., & Mattson, T. (2016). The TileDB array data storage manager. *Proceedings of the VLDB Endowment*, *10*(4), 349–360. https://doi.org/10.14778/3025111.3025117

Pebesma, E., Wagner, W., Schramm, M., Von Beringe, A., Paulik, C., Neteler, M., Reiche, J., Verbesselt, J., Dries, J., Goor, E., Mistelbauer, T., Briese, C., Notarnicola, C., Monsorno, R., Marin, C., Jacob, A., Kempeneers, P., & Soille, P. (2017). *OpenEO - a Common, Open Source Interface Between Earth Observation Data Infrastructures and Front-End Applications*. https://doi.org/10.5281/ZENODO.1065474

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

PricewaterhouseCoopers. (2019). *Copernicus market report*. Publications Office of the European Union. https://doi.org/10.2873/011961

Radiant Earth Foundation, & Contributors. (n.d.-a). *SpatioTemporal Asset Catalogs (STAC): Enabling online search and discovery of geospatial assets*. Retrieved April 19, 2021, from https://github.com/radiantearth/stac-spec

Radiant Earth Foundation, & Contributors. (n.d.-b). *STAC browser: A Vue-based STAC browser intended for static + dynamic deployment* [Computer software]. Retrieved May 29, 2021, from https://github.com/radiantearth/stac-browser

Rocklin, M. (2015). Dask: Parallel Computation with Blocked algorithms and Task Scheduling. *Proceedings of the 14th Python in Science Conference*, 126–132. https://doi.org/10.25080/majora-7b98e3ed-013

Rouse, J. W., Haas, R. H., Schell, J. A., Deering, D. W., & others. (1974). Monitoring vegetation systems in the Great Plains with ERTS. *NASA Special Publication*, *351*(1974), 309.

Roy, D. P., Li, J., Zhang, H. K., & Yan, L. (2016). Best practices for the reprojection and resampling of Sentinel-2 Multi Spectral Instrument Level 1C data. *Remote Sensing Letters*, *7*(11), 1023–1032. https://doi.org/10.1080/2150704x.2016.1212419

Rufin, P., Frantz, D., Yan, L., & Hostert, P. (2020). Operational Coregistration of the Sentinel-2A/B Image Archive Using Multitemporal Landsat Spectral Averages. *IEEE Geoscience and Remote Sensing Letters*, 1–5. https://doi.org/10.1109/lgrs.2020.2982245

Saha, B., & Srivastava, D. (2014). Data quality: The other face of Big Data. *2014 IEEE 30th International Conference on Data Engineering*, 1294–1297. https://doi.org/10.1109/icde.2014.6816764

Santos, L. A., Ferreira, K., Picoli, M., Camara, G., Zurita-Milla, R., & Augustijn, E.-W. (2021). Identifying Spatiotemporal Patterns in Land Use and Cover Samples from Satellite Image Time Series. *Remote Sensing*, *13*(5), 974. https://doi.org/10.3390/rs13050974

Schade, S., Granell, C., Vancauwenberghe, G., Keßler, C., Vandenbroucke, D., Masser, I., & Gould, M. (2019). Geospatial Information Infrastructures. In *Manual of Digital Earth* (pp. 161–190). Springer Singapore. https://doi.org/10.1007/978-981-32-9915-3_5

Scheffler, D., Frantz, D., & Segl, K. (2020). Spectral harmonization and red edge prediction of Landsat-8 to Sentinel-2 using land cover optimized multivariate regressors. *Remote Sensing of Environment*, *241*, 111723. https://doi.org/10.1016/j.rse.2020.111723

Schramm, M., Pebesma, E., Milenković, M., Foresta, L., Dries, J., Jacob, A., Wagner, W., Mohr, M., Neteler, M., Kadunc, M., Miksa, T., Kempeneers, P., Verbesselt, J., Gößwein, B., Navacchi, C., Lippens, S., & Reiche, J. (2021). The openEO API–Harmonising the Use of Earth Observation Cloud Services Using Virtual Data Cube Functionalities. *Remote Sensing*, *13*(6), 1125. https://doi.org/10.3390/rs13061125

Schuldt, B., Buras, A., Arend, M., Vitasse, Y., Beierkuhnlein, C., Damm, A., Gharun, M., Grams, T. E. E., Hauck, M., Hajek, P., Hartmann, H., Hiltbrunner, E., Hoch, G., Holloway-Phillips, M., Körner, C., Larysch, E., Lübbe, T., Nelson, D. B., Rammig, A., … Kahmen, A. (2020). A first assessment of the impact of the extreme 2018 summer drought on Central European forests. *Basic and Applied Ecology*, *45*, 86–103. https://doi.org/10.1016/j.baae.2020.04.003

Senf, C., Buras, A., Zang, C. S., Rammig, A., & Seidl, R. (2020). Excess forest mortality is consistently linked to drought across Europe. *Nature Communications*, *11*(1). https://doi.org/10.1038/s41467-020-19924-1

She, X., Zhang, L., Cen, Y., Wu, T., Huang, C., & Baig, M. (2015). Comparison of the Continuity of Vegetation Indices Derived from Landsat 8 OLI and Landsat 7 ETM+ Data among Different Vegetation Types. *Remote Sensing*, *7*(10), 13485–13506. https://doi.org/10.3390/rs71013485

Silver, A. (2017). Software simplified. *Nature*, *546*(7656), 173–174. https://doi.org/10.1038/546173a

Sinergise Ltd. (n.d.). *Sentinel Hub: Cloud API for Satellite Imagery.* Retrieved May 23, 2021, from https://www.sentinel-hub.com

Siqueira, A., Lewis, A., Thankappan, M., Szantoi, Z., Goryl, P., Labahn, S., Ross, J., Hosford, S., Mecklenburg, S., Tadono, T., Rosenqvist, A., & Lacey, J. (2019). CEOS Analysis Ready Data For Land – An Overview on the Current and Future Work. *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 5536–5537. https://doi.org/10.1109/IGARSS.2019.8899846

Sochat, V., & Contributors. (n.d.). *Singularity Python (spython): Streamlined Singularity Python client (spython) for Singularity.* [Computer software]. Retrieved May 12, 2021, from https://github.com/singularityhub/singularity-cli

Soille, P., Burger, A., Marchi, D. D., Kempeneers, P., Rodriguez, D., Syrris, V., & Vasilev, V. (2018). A versatile data-intensive computing platform for information retrieval from big geospatial data. *Future Generation Computer Systems*, *81*, 30–40. https://doi.org/10.1016/j.future.2017.11.007

Steele-Dunne, S. C., Friesen, J., & Giesen, N. van de. (2012). Using Diurnal Variation in Backscatter to Detect Vegetation Water Stress. *IEEE Transactions on Geoscience and Remote Sensing*, *50*(7), 2618–2629. https://doi.org/10.1109/tgrs.2012.2194156

Steinwand, D. R., Hutchinson, J. A., & Snyder, J. P. (1995). Map Projections for Global and Continental Data Sets and an Analysis of Pixel Distortion Caused by Reprojection. *Photogrammetric Engineering & Remote Sensing.* https://www.asprs.org/wp-content/uploads/pers/1995journal/dec/1995_dec_1487-1497.pdf

Strobl, P., Baumann, P., Adam, L., Zoltan, S., Brian, K., Matthew, P., Massimo, C., Stefano, N., Alex, H., & Trevor, D. (2017). The six faces of the data cube. *Proceedings of the 2017 conference on Big Data from Space (BIDS' 2017)*, 32–35. https://doi.org/10.2760/383579

Sudmanns, M., Tiede, D., Lang, S., Bergstedt, H., Trost, G., Augustin, H., Baraldi, A., & Blaschke, T. (2020). Big Earth data: disruptive changes in Earth observation data management and analysis? *International Journal of Digital Earth*, *13*, 832–850. https://doi.org/10.1080/17538947.2019.1585976

Sylabs, Inc. (n.d.). *Singularity Admin Guide: Installation on Windows or Mac.* Retrieved May 14, 2021, from https://sylabs.io/guides/3.7/admin-guide/installation.html#installation-on-windows-or-mac

Tanré, D., Deroo, C., Duhaut, P., Herman, M., Morcrette, J. J., Perbos, J., & Deschamps, P. Y. (1990). Technical note description of a computer code to simulate the satellite signal in the solar spectrum: The 5S code. *International Journal of Remote Sensing*, *11*(4), 659–668. https://doi.org/10.1080/01431169008955048

Tanré, D., Herman, M., Deschamps, P. Y., & Leffe, A. de. (1979). Atmospheric modeling for space measurements of ground reflectances, including bidirectional properties. *Applied Optics*, *18*(21), 3587. https://doi.org/10.1364/ao.18.003587

Thiel, C., & Schmullius, C. (2016). Comparison of UAV photograph-based and airborne lidar-based point clouds over forest from a forestry application perspective. *International Journal of Remote Sensing*, *38*(8-10), 2411–2426. https://doi.org/10.1080/01431161.2016.1225181

Thiel, M., Otte, I., Hill, S., Cluter, P., Förtsch, S., Sebold, S., & Löw, J. (n.d.). *eo2cube: Earth Observation Data Cubes of the University of Würzburg*. Department of Remote Sensing, University of Würzburg. Retrieved June 2, 2021, from https://datacube.remote-sensing.org/

Thomas, R., Cholia, S., Mohror, K., & Shalf, J. M. (2021). Interactive Supercomputing With Jupyter. *Computing in Science & Engineering*, *23*(2), 93–98. https://doi.org/10.1109/mcse.2021.3059037

Ticehurst, C., Zhou, Z.-S., Lehmann, E., Yuan, F., Thankappan, M., Rosenqvist, A., Lewis, B., & Paget, M. (2019). Building a SAR-Enabled Data Cube Capability in Australia Using SAR Analysis Ready Data. *Data*, *4*(3), 100. https://doi.org/10.3390/data4030100

TileDB, Inc. (n.d.). *TileDB Documentation*. Retrieved May 23, 2021, from https://docs.tiledb.com/main/

Truckenbrodt, J., & Contributors. (n.d.-a). *pyroSAR: A Python Framework for Large-Scale SAR Satellite Data Processing* [Computer software]. Retrieved May 8, 2021, from https://github.com/johntruckenbrodt/pyroSAR

Truckenbrodt, J., & Contributors. (n.d.-b). *pyroSAR Documentation: DEM Preparation*. Retrieved May 10, 2021, from https://pyrosar.readthedocs.io/en/latest/general/DEM.html

Truckenbrodt, J., Cremer, F., Baris, I., & Eberle, J. (2019). pyroSAR: A Framework for Large-Scale SAR Satellite Data Processing. In P. Soille, S. Loekken, & S. Albani (Eds.), *Proceedings of the 2019 conference on Big Data from Space (BiDS'19)* (pp. 197–200). Publications Office of the European Union. https://doi.org/10.2760/848593

Truckenbrodt, J., Freemantle, T., Williams, C., Jones, T., Small, D., Dubois, C., Thiel, C., Rossi, C., Syriou, A., & Giuliani, G. (2019). Towards Sentinel-1 SAR Analysis-Ready Data: A Best Practices Assessment on Preparing Backscatter Data for the Cube. *Data*, *4*(3), 93. https://doi.org/10.3390/data4030093

USGS. (2019, February 27). *Landsat WRS 2 Descending Path Row Shapefile*. U.S. Geological Survey. https://www.usgs.gov/media/files/landsat-wrs-2-descending-path-row-shapefile

Vermote, E. F., Tanre, D., Deuze, J. L., Herman, M., & Morcette, J.-J. (1997). Second Simulation of the Satellite Signal in the Solar Spectrum, 6S: an overview. *IEEE Transactions on Geoscience and Remote Sensing*, *35*(3), 675–686. https://doi.org/10.1109/36.581987

Villard, L., & Borderies, P. (2007). Backscattering Border Effects for Forests at C-band. *PIERS Online*, *3*(5), 731–735. https://doi.org/10.2529/PIERS061006120418

Vixie, P., Mašláňová, M., Dean, C., & Mráz, T. (n.d.). *cron(8) - Linux manual page* [Computer software]. https://man7.org/linux/man-pages/man8/cron.8.html

Warmerdam, F. (2008). The Geospatial Data Abstraction Library. In *Open Source Approaches in Spatial Data Handling* (pp. 87–104). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-74831-1_5

WEF. (2021). *The Global Risks Report 2021* [Research report]. The World Economic Forum; The World Economic Forum. https://www.weforum.org/reports/the-global-risks-report-2021

WEF. (2012). *Big Data, Big Impact: New Possibilities for International Development*. The World Economic Forum. http://www3.weforum.org/docs/WEF_TC_MFS_BigDataBigImpact_Briefing_2012.pdf

WEF, & Digital Earth Africa. (2021). *Unlocking the potential of Earth Observation to address Africa's critical challenges* [Research report]. The World Economic Forum. http://www3.weforum.org/docs/WEF_Digital_Earth_Africa_Unlocking_the_potential_of_Earth_Observation_to_address_Africa_2021.pdf

Wille, M., Clauss, K., Valgur, M., Sølvsteen, J., & Contributors. (n.d.). *Sentinelsat: Search and download Copernicus Sentinel satellite images* [Computer software]. Retrieved May 12, 2021, from https://github.com/sentinelsat/sentinelsat

Woodcock, R. (2019, July 18). *ODC Enhancement Proposal 002: Support for multiple DC databases*. Open Data Cube. https://github.com/opendatacube/datacube-core/wiki/ODC-EP-002---Support-for-multiple-DC-databases

Woodcock, R., & Kouzoubov, K. (2020, September 10). *Open Data Cube Core - GitHub Issue Nr. 1018*. https://github.com/opendatacube/datacube-core/issues/1018

Wulder, M. A., Loveland, T. R., Roy, D. P., Crawford, C. J., Masek, J. G., Woodcock, C. E., Allen, R. G., Anderson, M. C., Belward, A. S., Cohen, W. B., Dwyer, J., Erb, A., Gao, F., Griffiths, P., Helder, D., Hermosilla, T., Hipple, J. D., Hostert, P., Hughes, M. J., … Zhu, Z. (2019). Current status of Landsat program, science, and applications. *Remote Sensing of Environment*, *225*, 127–147. https://doi.org/10.1016/j.rse.2019.02.015

Wulder, M. A., Masek, J. G., Cohen, W. B., Loveland, T. R., & Woodcock, C. E. (2012). Opening the archive: How free data has enabled the science and monitoring promise of Landsat. *Remote Sensing of Environment*, *122*, 2–10. https://doi.org/10.1016/j.rse.2012.01.010

xcube Contributors. (n.d.). *xcube Documentation: Overview*. Retrieved June 2, 2021, from https://xcube.readthedocs.io/en/latest/overview.html

Yee, C., Durbin, C., Quinn, P., & Shum, D. (2020). *Task 51-Cloud-Optimized Format Study* (Technical Report No. 20200001178). NASA EOSDIS. https://ntrs.nasa.gov/citations/20200001178

Zhu, Z., & Woodcock, C. E. (2012). Object-based cloud and cloud shadow detection in Landsat imagery. *Remote Sensing of Environment*, *118*, 83–94. https://doi.org/10.1016/j.rse.2011.10.028

Zhu, Z., Wulder, M. A., Roy, D. P., Woodcock, C. E., Hansen, M. C., Radeloff, V. C., Healey, S. P., Schaaf, C., Hostert, P., Strobl, P., Pekel, J.-F., Lymburner, L., Pahlevan, N., & Scambos, T. A. (2019). Benefits of the free and open Landsat data policy. *Remote Sensing of Environment*, *224*, 382–385. https://doi.org/10.1016/j.rse.2019.02.016