HPC-oriented Canonical Workflows for Machine Learning Applications in Climate and Weather Prediction Postprint

Authors: Amirpasha, Mozaffari, Michael, Langguth, Bing, Gong, Jessica, Ahring, Adrian, Rojas Campos, Pascal, Nieters, Otoniel, José Campos Escobar, Martin, Wittenbrink, Peter, Baumann, Martin, G. Schultz, Amirpasha, Mozaffari

Date: 2022-11-28T00:00:00+00:00

Abstract

Machine learning (ML) applications in weather and climate are gaining momentum as big data and the

immense increase in High-performance computing (HPC) power are paving the way. Ensuring FAIR data and

reproducible ML practices are significant challenges for Earth system researchers. Even though the FAIR

principle is well known to many scientists, research communities are slow to adopt them. Canonical

Workflow Framework for Research (CWFR) provides a platform to ensure the FAIRness and reproducibility

of these practices without overwhelming researchers. This conceptual paper envisions a holistic CWFR

approach towards ML applications in weather and climate, focusing on HPC and big data. Specifically, we

discuss Fair Digital Object (FDO) and Research Object (RO) in the DeepRain project to achieve granular

reproducibility. DeepRain is a project that aims to improve precipitation forecast in Germany by using ML.

Our concept envisages the raster datacube to provide data harmonization and fast and scalable data access.

We suggest the Juypter notebook as a single reproducible experiment. In addition, we envision JuypterHub

as a scalable and distributed central platform that connects all these elements and the HPC resources to the

researchers via an easy-to-use graphical interface.

Full Text

Preamble

HPC-oriented Canonical Workflows for Machine Learning Applications in Climate and Weather Prediction

Amirpasha Mozaffari[1], Michael Langguth[1], Bing Gong[1], Jessica Ahring[1], Adrian Rojas Campos[2], Pascal Nieters[2], Otoniel José Campos Escobar[3], Martin Wittenbrink[4], Peter Baumann[3] & Martin G. Schultz[1]

[1]Forschungszentrum Jülich GmbH, 52425 Jülich, Germany
[2]Osnabrück University, 49074 Osnabrück, Germany
[3]Jacobs University Bremen, 28759 Bremen, Germany
[4]Deutscher Wetterdienst, 63067 Offenbach am Main, Germany

Keywords: FAIR; Reproducibility; Machine learning; Earth system sciences; Workflow

Citation: Mozaffari, A., et al.: HPC-oriented canonical workflows for machine learning applications in climate and weather prediction. Data Intelligence 4(2), 271-285 (2022). doi: 10.1162/dint_a_{00131}

Received: September 17, 2021; **Revised:** December 17, 2021; **Accepted:** February 5, 2022

Abstract

Machine learning (ML) applications in weather and climate are gaining momentum as big data and the immense increase in high-performance computing (HPC) power pave the way forward. Ensuring FAIR data and reproducible ML practices represents a significant challenge for Earth system researchers. Even though the FAIR principle is well known to many scientists, research communities have been slow to adopt them. The Canonical Workflow Framework for Research (CWFR) provides a platform to ensure the FAIRness and reproducibility of these practices without overwhelming researchers. This conceptual paper envisions a holistic CWFR approach toward ML applications in weather and climate, focusing on HPC and big data. Specifically, we discuss Fair Digital Object (FDO) and Research Object (RO) concepts in the DeepRain project to achieve

granular reproducibility. DeepRain aims to improve precipitation forecasting in Germany using ML.

Our concept envisages the raster datacube to provide data harmonization and fast, scalable data access. We suggest the Jupyter notebook as a single reproducible experiment and JupyterHub as a scalable, distributed central platform that connects all these elements and HPC resources to researchers via an easy-to-use graphical interface.

Corresponding author: Amirpasha Mozaffari (a.mozaffari@fz-juelich.de; ORCID: 0000-0001-6719-0425)

1. Introduction

The intention of the FAIR (Findable, Accessible, Interoperable, Reusable) principle by Wilkinson et al. [**?**] was not limited to data but also targeted other Digital Objects (DO) [**?**], such as algorithms, tools, and workflows that lead to data. The FAIR Digital Object (FDO) subsequently introduced by de Smedt et al. [**?**] provides a framework for transparent, reusable, and reproducible data [**?**]. The apparent benefit of reproducible science is that it becomes possible to restore results in critical situations, increase transparency, trust, interest, and the number of citations. It can rise to a level where reusing previous work becomes routine practice, leading to increased productivity, improved work habits, and greater continuity [**?**, **?**, **?**, **?**]. However, the reality of science deviates from these conveyed principles.

The concept of Canonical Workflow Framework for Research (CWFR) was proposed by Hardisty and Wittenburg [**?**] as a solution to expand the adaptation of the FAIR principle to the broader research community. CWFR relies on identifying recurring patterns across disciplines and breaking down workflows into smaller modular components that can be reused and reassembled for other use cases. In this paper, we suggest using Jupyter notebooks on JupyterHub with connection to an HPC system [**?**] as a platform to develop a concept for bringing the components of CWFR together for Machine Learning (ML) applications in Earth System Sciences (ESS).

To develop a functioning CWFR, identifying the challenges and practices of the particular community is essential. ESS in general, and climate and weather in particular, have seen significant growth in recent years thanks to petabyte-size data and exponential increases in computational capability [**?**, **?**]. With growing data volumes and in light of climate change and its impacts, FAIRness in ESS ensures comprehensible and reliable environmental knowledge. However, in the Earth sciences domain, more than 60% of surveyed researchers stated that they failed to reproduce someone else' s experiment, while over 40% admitted they were unable to reproduce their own experiment [**?**]. These issues also exist in ML, as documented in ML conference publications. At the prestigious Conference

on Neural Information Processing Systems (NIPS) in 2017, less than 40% of publications provided links to the code. Consequently, some studies highlight the importance of reproducible ML that allows others to apply contributions and increase the impact of ML research [**?**].

The data-driven nature of ML poses unique challenges regarding reproducibility. As more data is used for training and testing, ensuring that presented results are sound and reliable becomes increasingly difficult [**?**]. In addition, the training process involves randomness. For instance, stochastic gradient descent (widely used for ML model updates) employs a randomized procedure that could result in different weights at each run even when identical code is used [**?**]. Furthermore, ML frameworks are commonly used to speed up development. Mainstream frameworks such as TensorFlow [**?**] and PyTorch [**?**] use mixed precision for accelerating GPU training processes that could yield different results depending on underlying software or hardware. Most ML algorithms also use a vast range of libraries and frameworks, configurations, and virtual environments that rapidly change, so that different versions can lead to different outputs.

The challenges mentioned in ESS and ML compound when ML methods are applied to ESS data. Both ESS and ML algorithms rely on large data volumes that require rapid processing and thus high-performance computing (HPC) resources. Well-designed workflows play an important role in handling elaborate data preparation and efficiently utilizing tomorrow' s exascale computers. The widespread adaptation of workflow applications faces two significant challenges: a vast gap between workflow applications utilized in enterprise-scale IT firms and science labs, and shared beliefs that researchers' applications are unique [**?**]. More than 90% of researchers surveyed by Stoddart [**?**] agreed that "more robust experimental design" is necessary for enhanced reproducibility of scientific results. A well-designed workflow that ensures reproducibility and traceability at every step could greatly enhance the robustness of ESS experiment design. Having reusable software and workflow components does not imply that individual scientific ideas can no longer be pursued. By contrast, they will form a solid reproducible basis on which new ideas can build with less potential for errors.

Inspired by the expected benefits of integrating reproducibility in ML applied to ESS, we discuss the particular challenges in our interdisciplinary project, DeepRain, in Section 2. This project exemplifies the requirement for interoperable and reproducible research as it aims to improve precipitation forecasting using ML. In this context, the benefits of FAIRification on ML applications in ESS are discussed, and we describe how existing concepts and methods can be used in Section 3. Section 4 introduces our proposed framework based on CWFR, which aims to provide flexible and reproducible ML focusing on big data analysis on HPC systems. A conclusion and outlook based on our concept are given in the final Section 5.

2. Problem Formulation and Prerequisites

The DeepRain project serves as a concrete research example for which a canonical workflow framework is crucial. In DeepRain [**?**], more than 1.3 petabytes of meteorological data are exploited to develop complex ML algorithms to predict precipitation in Germany. Historical forecasts from the COSMO-DE (Consortium for Small-Scale Modelling) Ensemble Prediction System (EPS) provided by the German Weather Service (DWD) [**?**, **?**, **?**] serve as input for quantitative precipitation forecasts at station sites and on gridded domains based on tailor-made deep neural networks. For the latter, the high-resolution radar-based climatology product RADKLIM acts as a high-quality observational reference dataset [**?**, **?**].

Processing such large data volumes requires massive computational resources exclusively available on HPC systems. Due to bandwidth limitations, big data is usually hosted in the same facility that provides the HPC system to reduce data streaming delays and connection interruptions. Therefore, any suggested solution should provide easy access to HPC systems, stored data, and in-situ processing to avoid unnecessary uploads and downloads.

In the DeepRain project, a relatively broad group of researchers from ML, weather and climate, and computer science (CS) collaborate, requiring a framework that enables efficient collaboration without demanding extensive CS expertise. The project necessitates using technical tools, communicating and developing ML models and experiments, while reducing the time and energy needed to become acquainted with methods and terminology used across disciplines. FAIR practices help make research interoperable across disciplines, even within the same project and group.

One main obstacle to adopting FAIR practices is that many require drastic changes in researchers' procedures. Thus, any suggested solution should adjust to researchers' needs rather than being constrained by IT considerations [**?**]. Changes to researchers' work should be minimized to serve both the fulfillment of FAIR practices and researcher acceptance. In addition, any pre-designed workflow must allow the use of domain-specific language and procedures, for example with respect to result evaluation. Therefore, we narrowed our focus to ML applications for the ESS community to ensure easy adaptation without extensive development. If the proposed approach is adopted by the ESS community, it can be further developed and generalized to meet the requirements of other science communities.

A high degree of flexibility is necessary for any proposed CWFR to adapt to researcher practices. However, designing a flexible solution requires researchers to possess certain computer knowledge. Thus, researchers must be familiar with CS fundamentals such as Unix, bash, and version control (e.g., git). Much of the higher-level expertise needed to develop and maintain a CWFR is beyond the reach of typical research institutions. Therefore, collaboration between service providers, such as data and HPC centers, and research communities is necessary

to maintain a sustainable ecosystem. Such partnerships provide expertise, training, and infrastructure for research communities, requiring convergence between research communities and infrastructure providers.

Even though computational experiments should be easier to reproduce compared to physical experiments, the complexities and rapid pace of change in today' s software and hardware make it surprisingly difficult [**?**]. Research needs to be reproducible for humans (scientific reproducibility) and for machines (technical reproducibility). As mentioned above, randomized processes, mixed-precision, and hardware dependency impede technical reproducibility. Thus, we focus on scientific reproducibility, where we can reproduce statistical features and underlying distributions. This means that final results may deviate slightly from past experiments even with an identical setup, but the obtained statistical properties (e.g., model performance in terms of evaluation scores) and corresponding conclusions must remain the same. In this sense, deviations of obtained results must be indistinguishable from random noise.

---

3. FAIRness Building Blocks

We introduce components that can help build a FAIR ecosystem addressing the obstacles mentioned in Section 2 while reducing the cost of FAIRification.

As Kahn and Wilensky [**?**] introduced the concept of DO, it provides a framework to identify and trace any digital objects such as data, algorithms, and workflows. DOs, alongside Persistent Identifiers (PID) and metadata, constitute an FDO [**?**]. An FDO as a self-contained, typed, machine-actionable data package can provide basic components for a standardized, FAIR infrastructure [**?**]. FDO lays a foundation that can bring FAIRness to all components of science. As the adaptation of FDO is highly domain-oriented, we refer to Lannom et al. [**?**] for more details on applications for ESS.

Research Object (RO) is a related concept introduced by Bechhofer et al. [**?**] where the main focus is on born-digital objects and aggregation of data and collections. Thus, RO is well-suited for application in data-driven sciences. In addition, RO can be associated with DOI, making it findable and accessible over the internet. The RO concept relies on the idea that each RO provides a unit of knowledge. Therefore, RO acts as a container of resources (including a series of FDOs) and is shareable within and across different research groups. Our approach uses FDO and RO as building blocks to create a FAIR framework.

3.1 Notebook and Git

The emerging pattern in data-oriented research is to use in-situ analysis and visualization on HPC systems. This enables researchers to act quickly based on outputs, apply modifications, and restart workflows [**?**]. Jupyter notebook is a tool that researchers increasingly rely on [**?**]. Jupyter is also recognized

by Hardisty and Wittenburg [**?**] as one possible CWFR solution. Beg et al. [**?**] discuss the reproducibility of Jupyter notebooks as scientific workflows. Jupyter notebook provides a one-study, one-document concept that is easily shareable. In addition, it offers a user-friendly platform to document software while ensuring reproducibility by combining data, code, and software environment. While Jupyter notebook provides the core, JupyterHub expands the framework and brings flexibility to user groups [**?**]. JupyterHub provides access control and authentication, scalability with support for container and HPC technology, and portability from the cloud to local machines.

Despite these benefits, there are limitations to deploying notebooks as a CWFR solution. Any modification to the notebook is immediate, and multiple executions of a notebook with different inputs lead to loss of all previous information. In addition, any part of the notebook can be executed or skipped separately, creating a problem known as the undefined state of the notebook. Thus, the constant prototyping and rapid development ecosystem of Jupyter notebook threatens its adaptation as a reproducible workflow [**?**]. Changes applied to notebooks usually fall into two categories: 1) code developments to introduce new features and methods, or 2) experimenting in the hyperparameter space. The primary tool to track changes in algorithms remains version control, particularly git [**?**]. As many comprehensive studies address git' s application for reproducibility, we refer readers to Ram [**?**] rather than repeating that discussion. Each newly committed snippet of code is identified with a unique commit-ID.

ML applications often need to experiment with the parameter space, requiring intensive hyperparameter optimization and model space search. It is essential to preserve the notebook state and each experiment' s parameters. We propose an application of a notebook that we call an experiment dashboard, used to initiate any experiment notebook. Dashboard configuration, including a summary of all carried-out experiments and paths to data and the associated commit-ID for the current dashboard instance, is stored as FDOs. For executing individual notebook instances separately, we suggest a Python library called papermill that can run new instances with parameter values passed to them. Any new experiment with new parameter values is passed to a new notebook instance, and all are preserved. As shown in Figure 1 [Figure 1: see original paper], the experiment dashboard passes desired values to three independent notebook instances. Each instance is executed individually, and the instances can also run in parallel.

3.2 FDO and RO Modules

As there are no standard or widely used FDO libraries so far, a dedicated FDO module (FDOm) is envisioned to ensure proper FDO creation. We define principle rules for FDOs created by FDOm. Every unique experiment generates one FDO that points to one commit-ID generated by git. FDOs can either point to data or to another FDO; this referencing is called interlocking FDO. Any interlocking FDO appends all locked FDOs into one new FDO. Since FDOs

are only backward-looking, they can only link to data, commits, or other FDOs that exist at creation time. Besides, they include metadata and are searchable. We also introduce technical regulations: the FDO must provide information about the host system used, including system configuration data provided by the system admin and the environment, as well as libraries detected by FDOm.

For FDOm to be useful for ML applications, it is essential to document specific ML architecture and initial parameters. As TensorFlow [**?**] and PyTorch [**?**] are the main ML frameworks used in our project, FDOm will be tailored to receive the network summary directly. When an experiment ends, a unique PID is generated to identify the created FDO. FDOm stores this information as JSON-LD, which is human- and machine-readable. As shown in Figure 1, each notebook experiment instance is associated with a unique FDO.

Furthermore, we envision an RO module, called ROm, to create the necessary RO that may encapsulate several FDOs. Similar to FDOm, we foresee principle rules for ROm. Each RO has a state attribute that can be open, archived, or published. An open RO is mutable, and new FDOs can be added to it. An archived RO has been packaged and is therefore immutable. A published RO is similar to an archived one, except that a DOI is assigned to it. Any new RO can be created from scratch or based on an archived or published RO. The ROm allows researchers to inspect involved FDOs and select individual ones for later use. For example, any individual ML experiment can be selected and encapsulated as a new RO. We suggest preserving all experiments and their outputs regardless of whether they have been successful or not or whether they are intended for publication.

In contrast to FDO, the RO concept is already quite well-developed, and many implementations exist. From these implementations, we adapt our proposed ROm to be compatible with Ro-Crate [**?**]. We believe that the combination of FDOm and ROm, along with their integration with git and RO-Crate, could provide the necessary ecosystem to achieve high-level granular reproducibility.

3.3 Datacube Management

To address efficient data management challenges in the DeepRain project, we deployed an array-centric database. After evaluating several candidates, we opted for the Array Database Management System rasdaman [**?**], which offers geo-semantic query functionalities for multidimensional arrays, also referred to as datacubes [**?**]. In parallel to traditional file-based data, rasdaman provides efficient management of large-scale objects and standardized data modeling [**?**], contributing to data harmonization and enabling flexible access, extraction, analysis, and fusion of massive spatio-temporal datacubes based on a standardized query language. Figure 2 [Figure 2: see original paper] shows an exemplary query used to retrieve data from the DeepRain datacube. Data can be requested via query while the access pattern is registered in the related FDO.

4. The Proposed Framework

We propose a framework built on the tools mentioned previously or modified to fit research needs in ML applications in ESS. Our concept relies on a granular approach that uses FDO and RO as building blocks. Every single ML experiment is registered as an FDO. The method used to produce training data should be available to achieve reproducibility, complemented by explicit descriptions of the pipeline and architecture used. A series of experiments with corresponding FDOs is then encapsulated in an RO. This method ensures that the entirety of the research remains reproducible and that FAIRness is not limited to publication. Furthermore, to achieve granular FAIRness, we suggest a FAIR-test similar to a unit-test, a standard practice in CS where small code segments such as functions, methods, and classes are tested. The same principle can be integrated into research practices to validate the FAIRness of each research segment.

Our suggested CWFR is built around JupyterHub as a platform that glues notebooks, FDO, and RO together. The Jülich Supercomputing Centre (JSC) implemented an instance of JupyterHub called Jupyter-JSC[1] that provides a suitable platform for our proposed framework [**?**]. Jupyter-JSC has access to a wide range of data storage, CPU and GPU resources on JUWELS [**?**] and other HPC systems, and provides easily accessible integration with git.

The proposed framework is presented in the following prototype scenario. As researchers develop code, git provides a perfect avenue for preserving and reusing it in Jupyter notebook instances running on HPC infrastructure. When running ML experiments, the experiment dashboard initiates them. Required data is accessed via a path to data on the HPC system' s local file system, to an online repository, or via a query to the datacube interface. The experiment dashboard then creates several Jupyter notebook instances according to the number of experiments while passing ML experiment parameters. Each model calls FDOm, which sets up the associated FDO for all unique experiments, including data paths, ML architecture summaries, generated random seeds, etc. Relevant local copies of downloaded results are also stored and tracked. Any subsets of FDOs associated with ML experiments can be encapsulated as ROs with the help of ROm. The created RO can be used by researchers to share a holistic view of their work with collaborators, archive it, reuse it, or continue working based on previous achievements. Work has begun to implement this concept, and we plan to derive a more concrete scheme in a follow-up technical paper.

[1] https://jupyter-jsc.fz-juelich.de/

5. Conclusion and Future Work

We discussed necessary elements of a canonical workflow for ML applications in ESS to ensure FAIR and reproducible research while exploiting HPC capabilities. Our solution builds upon FDO and RO, and we envision a concept of FAIR unit-test where researchers can validate FAIR practices for small code segments and experiments. We introduced basic rules ensuring that FDO and RO are human- and machine-actionable and can achieve scientific reproducibility. For this, we proposed two modules—FDOm and ROm—to enforce these rules without introducing unnecessary changes to researcher workflows. The modules ensure each experiment is identified by a unique FDO and that series of experiments are encapsulated as ROs. In addition to file-based data storage, datacubes provide quick data access with an integrated FDO pointer function.

We proposed Jupyter notebook as the core of CWFR while acknowledging its limitation regarding the particular undefined state of a notebook. We suggest an experiment dashboard where researchers can initiate new experiments as independent notebooks. Papermill is a Python-based library that allows us to preserve and document changes in each notebook independently. The approach presented in this study aims to minimize technological barriers for ESS researchers to shift toward integrated FAIR practices. Nevertheless, elevating fundamental knowledge and skills in CS should remain a goal for ESS communities, because CS has developed many concepts and tools to ensure versioning, tracking, reproducibility, and portability. These tasks constitute the backbone of FAIR and reproducible research and are the pillars on which canonical workflows can be built.

References

[1] Wilkinson, M.D., et al.: Comment: The FAIR guiding principles for scientific data management and stewardship. Scientific Data 3, Article No. 160018 (2016)
[2] Kahn, R., Wilensky, R.: A framework for distributed digital object services. International Journal on Digital Libraries 6(2), 115–123 (2006)
[3] De Smedt, K., Koureas, D., Wittenburg, P.: FAIR digital objects for science: From data pieces to actionable knowledge units. Publications 8(2), Article No. 21 (2020)
[4] Lamprecht, A.-L., et al.: Towards FAIR principles for research software. Data Science 3(1), 37–59 (2019)
[5] Sandve, G. K., et al.: Ten simple rules for reproducible computational research. PLoS Computational Biology 9(10), e1003285 (2013)
[6] Gil, Y., et al.: Toward the geoscience paper of the future: Best practices for documenting and sharing research from data to software to provenance. Earth and Space Science 3(10), 388–415 (2016)
[7] Donoho, D.L.: An invitation to reproducible computational research.

Biostatistics 11(3), 385–388 (2010)
[8] Pineau, J., et al.: Improving reproducibility in machine learning research (A report from the NEURIPS 2019 Reproducibility Program). arXiv preprint arXiv: 2003.12206v2 (2020)
[9] Hardisty, A., Wittenburg, P.: Canonical Workflow Framework for Research (CWFR) - position paper - version 2 December 2020. Working paper. Available at: https://osf.io/9e3vc/. Accessed 9 December 2021
[10] Gobbert, J.H., et al.: Enabling interactive supercomputing at JSC: Lessons Learned. In: ISC High Performance 2018: High Performance Computing, pp. 669–677 (2018)
[11] Bauer, P., Thorpe, A., Brunet, G.: The quiet revolution of numerical weather prediction. Nature 525(7567), 47–55 (2015)
[12] Reichstein, M., et al.: Deep learning and process understanding for data-driven earth system science. Nature 566 (7743), 195–204 (2019)
[13] Stoddart, C.: Is there a reproducibility crisis in science? Nature (2016). Available at: https://doi.org/10.1038/d41586-019-00067-3. Accessed 9 December 2021
[14] Tatman, R., Vanderplas, J.: A practical taxonomy of reproducibility for machine learning research. In: The 2nd Reproducibility in Machine Learning Workshop at ICML 2018. Available at: https://hub.docker.com/r/pangwei/tf1.1/. Accessed 9 December 2021
[15] Beam, A.L., Manrai, A.K., Ghassemi, M.: Challenges to the reproducibility of machine learning models in health care. JAMA 323(4), 305–306 (2020)
[16] Developers, T.: TensorFlow. Available at: https://doi.org/10.5281/zenodo.5095721. Accessed 9 December 2021
[17] Paszke, A., et al.: PyTorch: An imperative style, high-performance deep learning library. In: The 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), pp. 1–12 (2019)
[18] DeepRain. Available at: https://www.deeprain-project.de/. Accessed 9 December 2021
[19] Doms, G., et al.: A description of the nonhydrostatic regional COSMO model. Part II: Physical parameterization, consortium for small scale modelling. Deutscher Wetterdienst. Available at: https://doi.org/10.5676/DWD_{pub}/nwv/cosmo-doc_5.00_{II}. Accessed 9 December 2021
[20] Gebhardt, C., et al.: Uncertainties in COSMO-DE precipitation forecasts introduced by model perturbations and variation of lateral boundaries. Atmospheric Research 100(2-3), 168–177 (2011)
[21] Peralta, C., et al.: Accounting for initial condition uncertainties in COSMO-DE-EPS. Journal of Geophysical Research: Atmospheres 117 (D7) (2012). Available at: https://doi.org/10.1029/2011JD016581. Accessed 9 December 2021
[22] Winterrath, T., et al.: Erstellung einer radargestützten Niederschlagsklimatologie. Available at: https://refubium.fu-berlin.de/handle/fub188/21892. Accessed 9 December 2021
[23] Winterrath, T., et al.: An overview of the new radar-based precipitation climatology of the Deutscher Wetterdienst—data, methods, products. In: Rainfall

Monitoring, Modelling and Forecasting in Urban Environment. UrbanRain18: The 11th International Workshop on Precipitation in Urban Areas, pp. 132–137 (2019)
[24] Wittenburg, P., Strawn, G.: Common patterns in revolutionary Infrastructures and data. EUDAT (2018). Available at: http://doi.org/10.23728/b2share.4e8ac36c0dd343da81fd9e83e728 Accessed 9 December 2021
[25] Ivie, P., Thain, D.: Reproducibility in scientific computing. ACM Computing Surveys 51(3), Article No. 63 (2019)
[26] Lannom, L., Koureas, D., Hardisty, A.R.: FAIR data and services in biodiversity science and geoscience. Data Intelligence 2(1-2), 122–130 (2020)
[27] Bechhofer, S., et al.: Research objects: Towards exchange and reuse of digital knowledge. Nature Precedings (2010). Available at: https://doi.org/10.1038/npre.2010.4626.1. Accessed 9 December 2021
[28] Weigel, T., et al.: Making data and workflows findable for machines. Data Intelligence 2(1-2), 40–46 (2020)
[29] Beg, M., et al.: Using Jupyter for reproducible scientific workflows. Computing in Science and Engineering 23(2), 36–46 (2021)
[30] Cholia, S., et al.: Towards interactive, reproducible analytics at scale on HPC system. In: Proceedings of UrgentHPC 2020: 2020 International Workshops on Urgent and Interactive HPC, pp. 47–54 (2020)
[31] Chacon, S., Straub, B.: Pro Git: Everything you need to know about Git, Apress. Second ed. Available at: https://git-scm.com/book/en/v2. Accessed 9 December 2021
[32] Ram, K.: Git can facilitate greater reproducibility and increased transparency in science. Source Code for Biology and Medicine 8(1), Article No. 7 (2013)
[33] Soiland-Reyes, S., et al.: Packaging research artefacts with RO-Crate. arXiv preprint arXiv: 2108.06503 (2021)
[34] Baumann, P., et al.: The multidimensional database system RasDaMan. SIGMOD Record 27, 575–577 (1998)
[35] Baumann, P., et al.: Array databases: Concepts, standards, implementations. Journal of Big Data 8, Article No. 28 (2021)
[36] Baumann, P.: The OGC Web coverage processing service (WCPS) standard. GeoInformatica 14, 447–479 (2010)
[37] Support, S.: JUWELS: Modular Tier-0/1 supercomputer at Jülich Supercomputing Centre. Journal of Large-scale Research Facilities 5, 1–8 (2019)

Author Biographies

Amirpasha Mozaffari is a postdoctoral researcher in the Earth System Data Exploration (ESDE) group at the Jülich Supercomputing Centre (JSC), part of Forschungszentrum Jülich. Trained as a geoscientist, he recently defended his Ph.D. in Computational Geohydrophysics from RWTH Aachen. He is active

in data management, workflow design, and FAIR data practices and co-chairs the canonical workflow framework for research in the Fair Digital Object forum. ORCID: 0000-0001-6719-0425

Michael Langguth received his Master' s degree in Meteorology in January 2016 from the Rheinische Friedrich-Wilhelms University of Bonn. During his Ph.D., he integrated and developed a hybrid parameterization scheme for convection into the ICON model, the operational weather prediction model by the German Weather Service DWD. In March 2020, he joined the Earth System Data Exploration (ESDE) group at Jülich Supercomputing Centre (JSC), where he develops neural networks for meteorological applications and conceptualizes workflows in the context of deep learning. ORCID: 0000-0003-3354-5333

Bing Gong received her Ph.D. from the Technical University of Madrid, Spain, in 2017. From 2017 to 2018, she worked as a Data Scientist at Corning Incorporated, Shanghai. Since January 2019, she has worked in the Earth System Data Exploration (ESDE) group at Jülich Supercomputing Center (JSC). Her current research interests are deep learning for earth science applications and machine learning software development. ORCID: 0000-0001-7770-2738

Jessica Ahring works in the Earth System Data Exploration (ESDE) group at Jülich Supercomputing Centre (JSC). She is studying for an MSc in Computer Science at RWTH Aachen with solid experience in database and HPC systems. ORCID: 0000-0002-1227-379X

Adrian Rojas Campos has been a Ph.D. student in Cognitive Science at Osnabrück University, Germany, since 2019. He obtained his BSc in Psychology in 2017 and his MSc in Cognitive Science from Osnabrück University in 2019. His area of interest is the integration of deep learning algorithms in the research processes of multiple sciences. ORCID: 0000-0002-9036-1031

Pascal Nieters is a Ph.D. student in the Neuroinformatics Lab at the Institute for Cognitive Science at Osnabrück University, where he works on computational models of neural computation. He develops models that help understand how neural tissue in brains physically solves computational problems and how this understanding may be transferred to develop new computer hardware. He also applies neural network models as a machine learning tool to help understand and predict the behavior of dynamical systems. ORCID: 0000-0003-0538-6670

Otoniel José Campos Escobar is a Ph.D. candidate at Jacobs University, Bremen. His research focuses on the integration of linear algebra in array databases for machine learning applications. He has worked as a researcher and software engineer in the DeepRain project since July 2020 on integrating weather data and datacubes using the array database rasdaman. ORCID: 0000-0003-0656-3747

Martin Wittenbrink is a Scientist at the German Meteorological Service DWD. He earned his Master of Science degree in Physics of the Earth and Atmosphere at the University of Bonn in 2020. He is currently part of the research

project DeepRain and develops spatial precipitation postprocessing methods. ORCID: 0000-0002-6227-7609

Dr. Peter Baumann is Professor of Computer Science, inventor, and entrepreneur. He received his Ph.D. with "summa cum laude" from Technical University Darmstadt, Germany, in 1993. At Jacobs University, Bremen, Germany, he researches flexible services for massive multi-dimensional datacubes and their application in science and engineering. In this field, he has published over 160 book chapters, journal, and conference articles and has internationally patented the datacube technology. With the rasdaman Array Database system, proven on Petabyte-scale spatio-temporal databases and across thousands of cloud nodes, he pioneered the field of array databases and datacubes. For its successful commercialization, he founded and led hitech spinoff Rasdaman GmbH. Rasdaman has received a series of international innovation awards, such as the 2019 US TechConnect Award and the 2019 DIN Innovator Award. For many years, Peter Baumann has critically shaped and often led datacube standards in ISO, OGC, and the European legal framework for a common SDI, INSPIRE. See www.peter-baumann.org for more information. ORCID: 0000-0003-3860-4726

PD Dr. Martin Schultz works at the interface between atmospheric and computer science. He obtained his Ph.D. at Forschungszentrum Jülich in 1995 and worked at Harvard University and the Max Planck Institute for Meteorology before returning to Jülich in 2006. Since 2017, he has established the research group on Earth System Data Exploration (ESDE), which develops new machine learning methods and FAIR data workflows for Earth system science at the Jülich Supercomputing Centre (JSC). ORCID: 0000-0003-3455-774X

Note: Figure translations are in progress. See original paper for figures.