album: a framework for scientific data processing with software solutions of heterogeneous tools

Jan Philipp Albrecht\textsuperscript{1,2} Deborah Schmidt\textsuperscript{1,2} Kyle Harrington\textsuperscript{1,3}

\textsuperscript{1} - Max Delbrück Center for Molecular Medicine in the Helmholtz Association, Berlin, Germany
\textsuperscript{2} - co-first authors
\textsuperscript{3} - corresponding author: mdc@kyleharrington.com

Summary

album (Schmidt, Albrecht, and Harrington 2021) is a decentralized distribution platform for solutions to specific scientific problems. It works across platforms, tools, and data domains and is designed to address limitations in reproducibility of scientific data software solutions and workflows, particularly when interactivity is needed. album can be used to programatically define how to interoperate between applications. It can ship versatile applications while tweaking them for a specific target audience or use case. An updated list of features and applications can be found on the documentation site.

Composition

A schema of the composition of all album components is shown in figure Figure 1. album is deployed as a pip package for command line usage. Each album solution can be defined using a single Python file, including metadata like authorship, usage instructions, and a description of the compatible Conda (Anaconda, Inc. 2021) environment. The metadata format is derived from the bioimage.io specification (Beuttenmüller, F. and Pape, C. and Ouyang, W. and Kuttra, D. and Melnikov, E. and Schmidt, D. and Gómez de Mariscal, E. and Novikov, M. 2021). Solutions can be shared via URL, hard drive location, or deployed to zenodo and shared via DOI. Catalogs make it possible to bundle solutions and also share them via git repository.

Moreover, we have created a graphical user interface based on SciJava (Rueden et al. 2017) called album-app (Schmidt 2021). It guides the user through setting
Figure 1: The components of the album ecosystem are shown. album is the core tool for accessing catalogs and solutions. Remote catalogs (shown in gray) can be added and removed to a user’s local collection which enables solutions to be installed, tested, and run. album-app is a graphical interface to album that provides a user-friendly way to access catalogs and run solutions.
up the entire album installation and makes it easy to add catalogs, install and run solutions, and handling input parameters graphically. An example screenshot of album-app is shown in Figure 2.

![Figure 2: An example screenshot of the album-app user interface displaying the user’s local collection dialog. The interface provides a way to quickly add new album catalogs, load existing catalogs into the collection, and quickly run recently installed solutions.](image)

**Statement of need**

There is a vast spectrum of available scientific (image analysis) tools with novel approaches appearing online on a daily basis. However, many are hard to install, have nonexistent, or outdated documentation, are unmaintained, and error prone. Often scientists are neither trained nor encouraged to focus on software reproducibility, usability, maintenance, and interoperability. When trying to solve a specific problem it becomes increasingly unfeasible to review all existing tools and plugins. album is aiming for a different view by helping scientists to describe use cases and automate solutions as much as possible to allow reproducibly sharing specific results (see figure Figure 3).

Existing package managers like Conda (Anaconda, Inc. 2021), npm (npm, Inc. 2021), apt (The Debian Project 2020), or Homebrew (Max Howell 2021) can be used by non experts, but they don’t provide scientists with a well defined, reproducible entry point for solving a specific problem. Some lack cross platform support or a way to deploy packages to a self hosted location. album runs on Linux, Windows and MacOS. album solutions can be collectively shared via so called catalogs, living on a hard drive location or a git repository. album provides
Figure 3: album focuses on automating tool usage based on specific scientific problems. Currently software tools that are designed to solve problems have independent interfaces for execution. album provides a unified method for installing and running solutions in addition to a catalog that collects software solutions into a single location.

all the functionalities to enable research software facilities (e.g. imaging facilities) to deploy a curated list of software tools in an easily maintainable fashion.

Existing scientific software frameworks like Fiji (Schindelin et al. 2012), ImJoy (Ouyang et al. 2019), and napari (Sofroniew, N., Talley Lambert, Evans, K., Nunez-Iglesias, J., Yamauchi, K., Solak, A. C., et al. 2020) are often limited in language support and bound to the environment of the base software. Many users therefore suffer from version clashes or lack of version transparency. In album each solution defines it’s own Conda environment, making it possible to launch most open source tools from a simple python script snippet included in the solution and adjust it’s usage based on the problem to be solved. The environments can and should include fixed versioning to be as reproducible as possible. We currently provide templates for Python, Java, ImageJ, ImageJ2 (Rueden et al. 2017), BigDataViewer (Pietzsch et al. 2015) and ImgLib2 (Pietzsch et al. 2012), sciview (Günther and Harrington 2020), and napari (Sofroniew, N., Talley Lambert, Evans, K., Nunez-Iglesias, J., Yamauchi, K., Solak, A. C., et al. 2020).

Existing workflow tools like biaflows (Rubens et al. 2020), Snakemake (Köster and Rahmann 2012), Galaxy (Goecks, Nekrutenko, and Taylor 2010), or Nextflow (Di Tommaso et al. 2017) aimed to solve scientific problems in a reproducible manner have a high entry barrier and are therefore rarely used by people without a computer science degree. While user-friendly workflow tools like KNIME (Berthold et al. 2009) often lack the ability to run native applications and focus on cloud computing instead. album provides a user-friendly way to
develop and distribute reproducible scientific solutions that can even include interactive, and native applications. We provide graphical interfaces to setup and launch solutions. album can also run multiple solutions in a row - this feature will be expanded in the future.

Connecting existing lab notebook tools like LabBook (Kandogan et al. 2015) with album could enable scientists to log scientific procedures conveniently by making sure computational setups can easily be reproduced by other labs.

Acknowledgements
We acknowledge support from the Helmholtz Imaging.

References
Anaconda, Inc. 2021. Conda (version 4.10.1). https://conda.io.

Berthold, Michael R, Nicolas Cebron, Fabian Dill, Thomas R Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. 2009. “KNIME—the Konstanz Information Miner: Version 2.0 and Beyond.” ACM SIGKDD Explorations Newsletter 11 (1): 26–31.

Beuttenmüller, F. and Pape, C. and Ouyang, W. and Kutra, D. and Melnikov, E. and Schmidt, D. and Gómez de Mariscal, E. and Novikov, M. 2021. https://github.com/bioimage-io/spec-bioimage-io/.

Di Tommaso, Paolo, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. 2017. “Nextflow Enables Reproducible Computational Workflows.” Nature Biotechnology 35 (4): 316–19.

Goecks, Jeremy, Anton Nekrutenko, and James Taylor. 2010. “Galaxy: A Comprehensive Approach for Supporting Accessible, Reproducible, and Transparent Computational Research in the Life Sciences.” Genome Biology 11 (8): 1–13.

Günther, Ulrik, and Kyle IS Harrington. 2020. “Tales from the Trenches: Developing Sciview, a New 3d Viewer for the ImageJ Community.” In Eurographics: VisGap - the Gap Between Visualization Research and Visualization Software.

Kandogan, Eser, Mary Roth, Peter Schwarz, Joshua Hui, Ignacio Terrizzano, Christina Christodoulakis, and Renée J Miller. 2015. “Labbook: Metadata-Driven Social Collaborative Data Analysis.” In 2015 IEEE International Conference on Big Data (Big Data), 431–40. IEEE.

Köster, Johannes, and Sven Rahmann. 2012. “Snakemake—a Scalable Bioinformatics Workflow Engine.” Bioinformatics 28 (19): 2520–22.

Max Howell. 2021. Homebrew (version f7689d1). https://brew.sh.
Ouyang, Wei, Florian Mueller, Martin Hjelmare, Emma Lundberg, and Christophe Zimmer. 2019. “ImJoy: An Open-Source Computational Platform for the Deep Learning Era.” Nature Methods 16 (12): 1199–1200.

Pietzsch, Tobias, Stephan Preibisch, Pavel Tomančák, and Stephan Saalfeld. 2012. “ImgLib2—Generic Image Processing in Java.” Bioinformatics 28 (22): 3009–11.

Pietzsch, Tobias, Stephan Saalfeld, Stephan Preibisch, and Pavel Tomancak. 2015. “BigDataViewer: Visualization and Processing for Large Image Data Sets.” Nature Methods 12 (6): 481–83.

Rubens, Ulysse, Romain Mormont, Lassi Paavolainen, Volker Bäcker, Benjamin Pavie, Leandro A Scholz, Gino Michiels, et al. 2020. “BIAFLOWS: A Collaborative Framework to Reproducibly Deploy and Benchmark Bioimage Analysis Workflows.” Patterns 1 (3): 100040.

Rueden, Curtis T, Johannes Schindelin, Mark C Hiner, Barry E DeZonia, Alison E Walter, Ellen T Arena, and Kevin W Eliceiri. 2017. “ImageJ2: ImageJ for the Next Generation of Scientific Image Data.” BMC Bioinformatics 18 (1): 1–26.

Schindelin, Johannes, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, et al. 2012. “Fiji: An Open-Source Platform for Biological-Image Analysis.” Nature Methods 9 (7): 676–82.

Schmidt, Deborah. 2021. Album-App (version ab0bb99b). https://gitlab.com/album-app/album-app.

Schmidt, Deborah, Jan Philipp Albrecht, and Kyle Harrington. 2021. Album (version 21a329f4). https://album.solutions.

Sofroniew, N., Talley Lambert, Evans, K., Nunez-Iglesias, J., Yamauchi, K., Solak, A. C., et al. 2020. https://zenodo.org/record/4048613.

The Debian Project. 2020. Advanced Package Tool (Apt) (version 1.8.2.2). https://wiki.debian.org/Apt.