The DataLad Handbook

A flexible, extendable & reusable open source framework for user-focused and workflow-driven software documentation

Wagner, A. S.[1] , Waite, L. K.[1], Waite, A.Q.[1], Reuter, N.[1] Poldrack, B.[1], Poline, J-B.[3], Kadelka, T.[1], Markiewicz, C. J.[4], Vavra, P.[5], Paas, L. K.[1], Herholz, P.[3], Mochalski, L.[1], Wiersch, L.[1], Kraljevic, N.[1], Heckner, M.[1], Chormai, P.[6], Halchenko, Y.O.[7] & Hanke, M.[1,2]

[1]Institute of Neuroscience and Medicine, Juelich Research Centre; [2]Institute of Systems Neuroscience, Heinrich Heine University Düsseldorf; [3]McGill University, Montreal; [4]Stanford University, Stanford; [5]Department of Biological Psychology, OvGU Magdeburg, [6]Max Planck School of Cognition, Leipzig; [7]Dartmouth College, Hanover (NH);

Background

DataLad (Halchenko, Hanke et al.) is a comprehensive data management tool and can help to solve various data management problems such as file size independent version control, data sharing, data storage and backup, computationally reproducible data analysis, or metadata management.

But: The functionality of any tool remains inaccessible or unknown if it is not sufficiently documented.

Complex software needs accessible user documentation!

The DataLad Handbook is a comprehensive documentational resource that fulfills the needs of different software **user types** independent of background:

- **Trainees** learn the tool
- **Planners** assess applicability of the tool
- **Teachers** teach how the tool is used

Book structure

Introduction

General overview: What can users expect from software & handbook

Guidance: Usecase-dependent chapter suggestions

Installation: Instructions for all common infra-structures

Basics

Domain-agnostic tutorial: Narrative-based code-along course with tested code snippets

Basic software skills: Provides a broad exploration of the software in a continuous, project-based workflow

Trial and error: Common errors are explicitly demonstrated in the safe-space of a tutorial

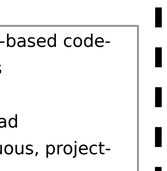
Optional advanced information: Toggleable or custom sections contain extra information. This keeps the visible information consise, but allows for exploration of advanced contents

Advanced

Documentation for advanced users: Non-technical demonstration of advanced features

Extension documentation: Workflow-based glimpses into domain-specific additions to the DataLad ecosystem

Use cases



- **Common uses showcase:** Non-technical descriptions of real-world applications.
- **Step-by-step recipes:** Explanations and instructions towards the described real-world application. Links to required chapters where needed.

Read the Handbook!
Learn or or teach about DataLad with the book at **handbook.datalad.org**!

Technical infrastructure

- Flexible, extendable & reusable open source infrastructure: Python-based, written in RST markup using Sphinx, hosted on readthedocs.org, illustrations by undraw.io, source code on GitHub, continuous integration using Travis & readthedocs.org
- Multiple formats: HTML, PDF, EPUB
- Custom Sphinx extensions (github.com/mih/autorunrecord) allow code execution and record code output. The handbook in itself is a framework for workflow testing.
- Simple audience tailoring: Different branches or tags can be rendered simultaneously, allowing dedicated URLs for different content. Example: Stand-alone branch for institute-internal workflows
- CC-BY-SA: Feel free to use the handbook infrastructure for your documentation project, e.g. Princeton Handbook for Reproducible Neuroimaging (Brooks et. al)

Conjunct software & user-documentation development

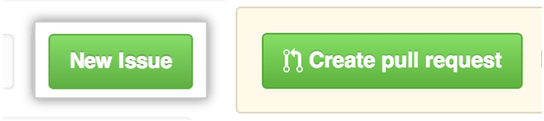
Advantages

- Higher rates of bug detection
- User-based documentation efforts uncover deficiencies of technical docs and user experience
- Workflow-based demos highlight API inconsistencies
- Documentation challenges facilitate software development

Caveats

- User-documentation does not replace technical docs
- Premature feature documentation: helpful for feedback & software dev facilitation, but increases documentation workload
- Separate software and user-docs rely on synchronized release management. Otherwise, unreleased functionality is documented publicly

Contributing



- GitHub-based development allows different contribution types
- Low-barrier contributions: General improvements, feature requests, feedback. High-barrier contributions (for advanced users): Content contributions, technical infrastructure
- "Basics/Advanced": Discussions on order/emphasis, feature requests
- "Use cases": Users contribute their DataLad workflows
- Technical infrastructure and visuals: Contributions to artwork dataset or handbook support software

All contributions are reviewed by the DataLad core developer team

Community and acknowledgement

- Credit is given for commit- and not commit-based contributions
- Co-authorship (PDF/EPUB + each Zenodo release), recognition with allcontributors-bot (allcontributors.org, following The Turing Way project; the-turing-way.netlify.org)
- Future directions: Presence in Hackathons/Hacktoberfest/ etc.
- Goal: Users share their individual workflows as use cases
- Current contributor count: 26

Want to learn more?
Find the source code on GitHub

Contribute!
Issues, PRs, or feature/topic requests are always welcome!

Heinrich Heine Universität Düsseldorf

1129855
1429999

SPONSORED BY THE
Federal Ministry of Education and Research
01GQ1112
01GQ1411

References

Brooks, P. P., et al. (2020, February 26). Princeton Handbook for Reproducible Neuroimaging (Version v0.1.0). Zenodo. http://doi.org/10.5281/zenodo.3688789

Hanke, M., Halchenko, Y. O. et al. (2020, May 22). datalad/datalad (Version 0.13.0rc2). Zenodo. http://doi.org/10.5281/zenodo.3840589