

CWL Viewer: The Common Workflow Language Viewer

Mark Robinson¹, Stian-Soiland-Reyes¹, Michael Crusoe², Carole Goble¹

¹ The University of Manchester, UK. Email: soiland-reyes@manchester.ac.uk

² Common Workflow Language project

Project Website: <https://view.commonwl.org>

Source Code: <http://github.com/common-workflow-language/cwlviewer>

License: CWL Viewer is licensed under the terms of the Apache License, Version 2.0, see <https://www.apache.org/licenses/LICENSE-2.0>

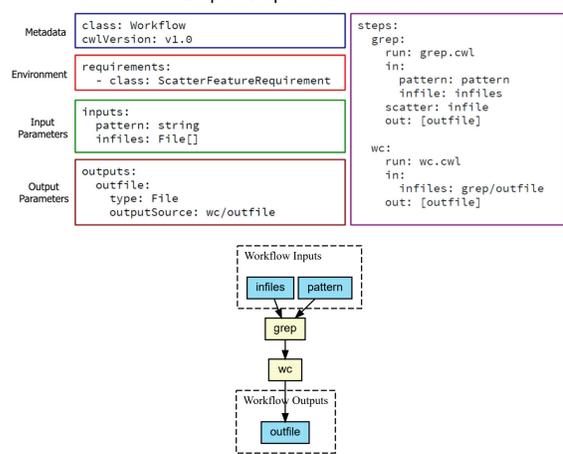
Abstract

Facilitating easy browsing, sharing and understanding of scientific workflows written in the Common Workflow Language

The *Common Workflow Language* (CWL) project emerged from the BOSC 2014 Codefest as a grassroots, multi-vendor working group to tackle the portability of data analysis workflows. Its specification for describing workflows and command line tools aims to make them portable and scalable across a variety of computing platforms.

At its heart CWL is a set of **structured text** files (YAML) with various **extensibility** points to the format. However, the CWL **syntax** and multi-file collections are not conducive to workflow **browsing, exchange and understanding**: for this we need a **visualization suite**.

Example: Grep and Count Workflow



CWL Viewer is a richly featured **web visualization suite** which graphically presents and lists the details of CWL workflows with their **inputs, outputs and steps**. It also packages the CWL files into a downloadable **Research Object Bundle** including **attribution, versioning and dependency** metadata in the manifest, allowing it to be easily shared.

The tool operates over **any workflow** held in a GitHub repository. Other **features include**: **path visualization** from parent and child nodes; **nested workflows** support; **workflow diagram download** in a range of image formats; a **gallery** of previously submitted workflows; and support for private git repositories and public GitHub including live updates

Permalinks and Archiving

The workflow pages contain the full GitHub URL to the workflow within a repository, enabling them to be easily remembered and manipulated within the browser.

<https://view.commonwl.org/workflows/github.com/common-workflow-language/workflows/blob/master/workflows/lobSTR/lobSTR-workflow.cwl>

GitHub URL

If a Git commit ID is used when adding the workflow, the visualisation and download become a **"snapshot"** of the workflow at a moment in time and exist at that **link permanently** for sharing.

Otherwise the visualisation will track the **branch** of the Github repository for updates after a cache expires.

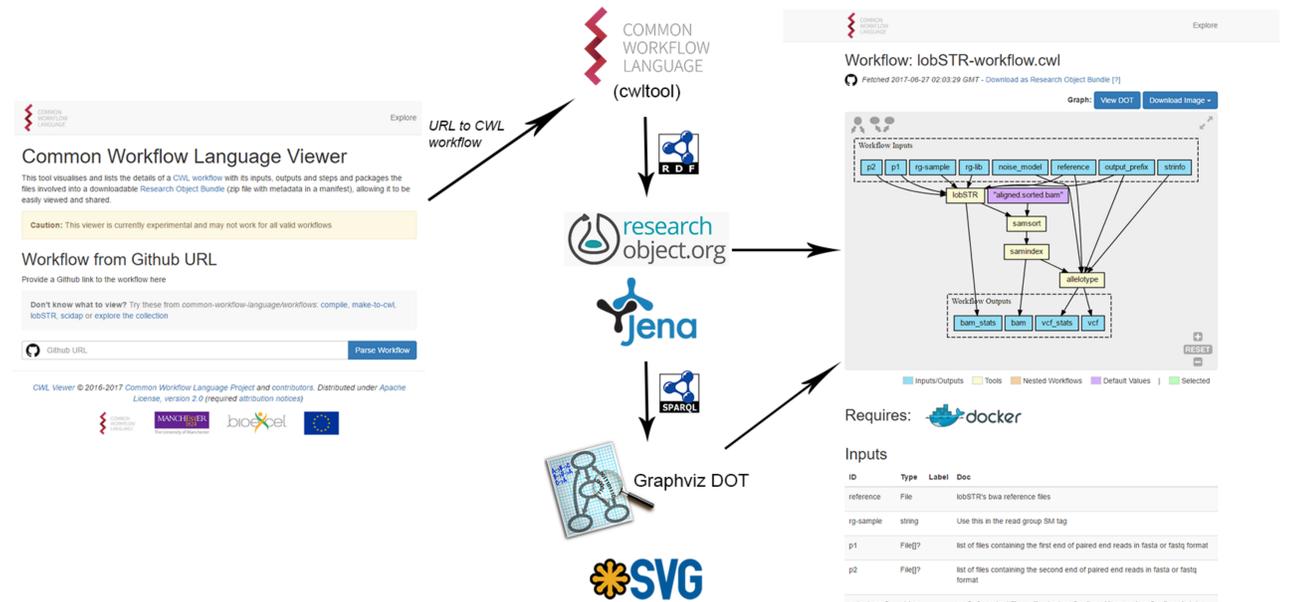
Provenance



CWL Viewer captures the **git commit log** and converts it to W3C **PROV** statements, which are added to the **Research Object** manifest. This provides **authorship and versioned permalinks** for each constituent part of the CWL workflow.

Work supported by BioExcel, funded by the European Union Horizon 2020 program under grant agreement 675728

Application Overview

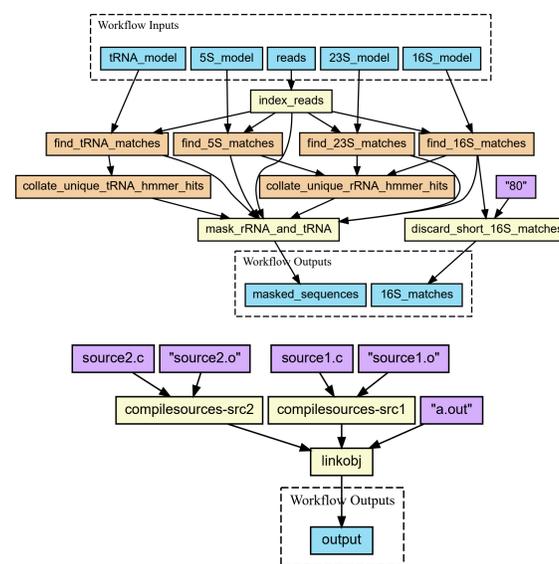


Visualisation

Visualisations allow complex workflows to be understood quickly and easily even without knowledge of the Common Workflow Language syntax

Directed acyclic graphs are created from the workflow descriptions provided using *Graphviz DOT*. The source for this is provided within the application and can be downloaded for use in external applications.

The diagram can be **downloaded** in various formats and included in reports, presentations and external documentation to illustrate and explain their function.

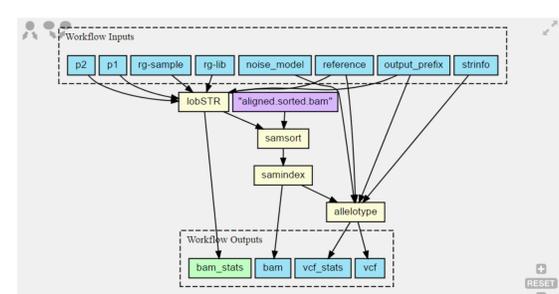


Colours differentiate **inputs and outputs** (blue), **tool steps** (yellow), **nested workflows** (orange) and **default values** (purple).

Labels within the workflow description are included instead of the ID of the element if provided (as well as documentation strings being included in the detailed tables underneath on the page). This encourages **best practices** when writing CWL workflows.

Within the web application the visualisation can also be **panned** and **zoomed** to navigate very large workflows.

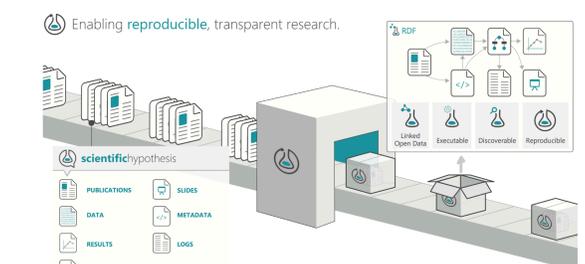
Steps can also be selected in green, which highlights the corresponding row in the table included on the page.



After this, the **selection can be expanded to parent or child nodes** by clicking the corresponding buttons. This helps to quickly distinguish parts of the workflow contributing to particular elements, eg an output.

Sharing and Research Objects

Research Objects (RO) are an emerging approach to publishing scholarly information on the web and support **reuse and reproducibility**. They consist of a *container* of files with a **manifest** to provide meaningful information about what those files *are*, what they *mean*, how they relate and provide **provenance and versioning** information.



The *Common Workflow Language* is not designed to capture detailed Description, provenance and versioning information within a workflow description. However, together the two technologies are a **natural combination** to provide both the language and packaging elements of execution and produce a **self-contained** workflow which can easily be run and reasoned about when published online.



CWLViewer produces a download of each workflow added in the form of a **Research Object Bundle** (.zip container) in order to facilitate sharing and reuse. *Zenodo* can also be used with this to obtain a DOI.

Discovery

CWL Viewer also provides a **gallery** of workflows which have been previously viewed using the site. A basic overview of the workflow, *where* it is from and a **thumbnail** of the visualisation are given.

This allows users to **locate** workflows which are relevant to them and provides a bank of workflows to **learn** and see helpful techniques being utilised within the Common Workflow Language. It also helps to avoid repetition by the community when developing workflow.