



FAIR hackathon 2021

Session 3: Workflow & Tools FAIRification

Carole Goble
Stian Soiland-Reyes

Finn Bacall
Stuart Owen
Douglas Lowe
Nick Juty
Simone Leo

EOSC-Life WP3 FAIRification workshop
2021-12-16



Schedule

- Overview of workflows and workflow management software
- Brief Mentimeter survey of participants
- Making my Workflows FAIR
- Incorporating my tool and making it FAIR Workflow ready
- Final discussion

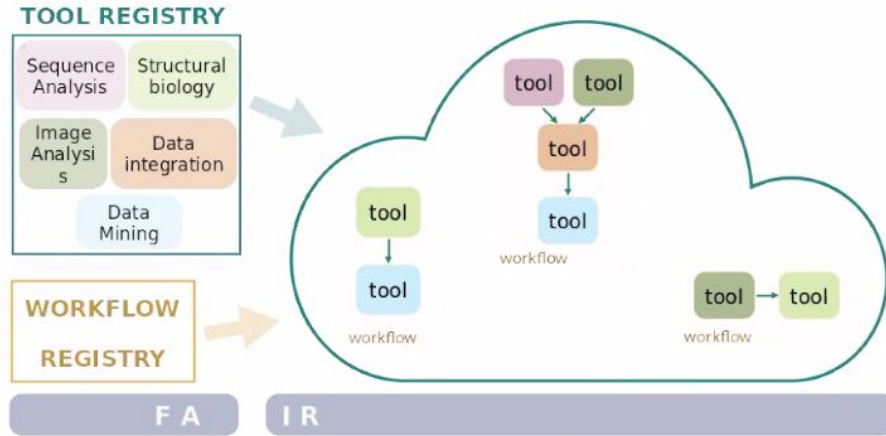
Please help take [collaborative notes](#)!



A data and method commons



A portable environment of interoperable tools



Workflows are an entry point to the tools and datasets

functions for production quality FAIR data processing

access to secure data processing

RIs publish data, methods & services for management, storage and reuse

Figure Credit: Romain Dallet

Galaxy Genome Annotation (GGA) environment in the cloud

Mentimeter questions for the participants



Do you already use workflows?

If yes, which systems?

If no, which systems do you intend to use?

Do you have tools that need to be wrapped to run in workflows?

If so, are they command line, web services, containers?

Will you be using sensitive data?

Will you need to operate in a Trusted Research Environment?

What kind of provenance do you need to collect?

- data lineage (traceability of data)
- workflow execution (recomputability of data)

Is computational portability important?

Do you have test cases set up for your tools / workflows?





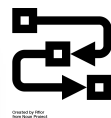
What is a Workflow?

Why use a Workflow Management System

**What kind of systems are available for
what kind of problems?**



Workflows are an entry point to Computational Science and a way to run & share multi-step methods



Multi-step (semi)-automated data processing pipelines, simulation studies and analytics



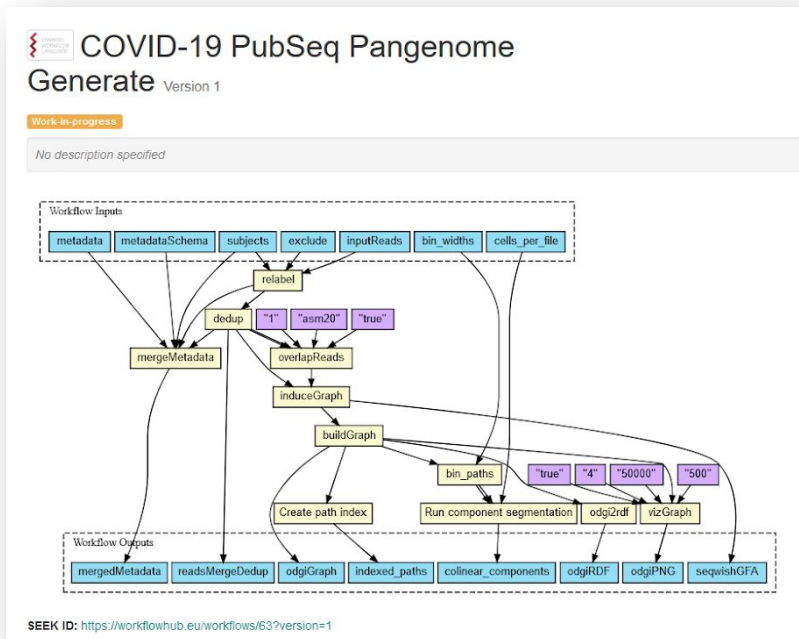
Access to (distributed) data, tools; Scalable use of computational resources



Simplified tool chains, auto-documentation, shielded from technical heroics



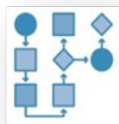
Shared, validated and comparative know-how



Workflow specification & execution & tools



Specification description



Workflow



Parameters



Inputs
Outputs



Guidance



Software Execution



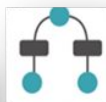
WfMS
Engine



Associated Objects



Data



Logs /
Histories /
Provenance



Contextual Entities
Metadata Graphs



Sample input
parameters, test data



Services,
e.g. Test engines



Related workflows
Checker workflows



Why are Workflow Systems Useful



Abstraction & Composition



- Using the best codes written by 3rd parties
- Handle heterogeneity
- Shield complexity & incompatibility
- Sharable reusable, re-mixable methods



Sharing & Adaptability



- Shared method, publishable know-how
- BYOD / parameters
- Different implementations
- Changes in execution infrastructure

Automation



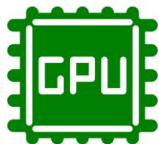
- Repetitive reproducible pipelines
- Simulation sweeps
- Manage data and control flow
- Optimised monitoring & recovery
- Automated deployment

Reporting & Accreditation



- Provenance logging & data lineage
- Auto-documentation
- Result comparison

Scalability & Infrastructure Access



- Accessing infrastructures, datasets and tools
- Optimised computation and data handling
- Parallelisation
- Secure sensitive data access & management
- Interoperating datasets & permission handling

Portability



- Dependency handling
- Containerisation & packaging
- Moving between on premise & cloud



What benefits to me does a Workflow Management System offer?



Heavy lifting

- The WfMS handles the tricky infrastructure bits and dependencies

Documented method

- Process, debug and record
- Publishable results

There is an upfront cost for downstream benefits

Benefits are best when a community buys in and workflows are supported

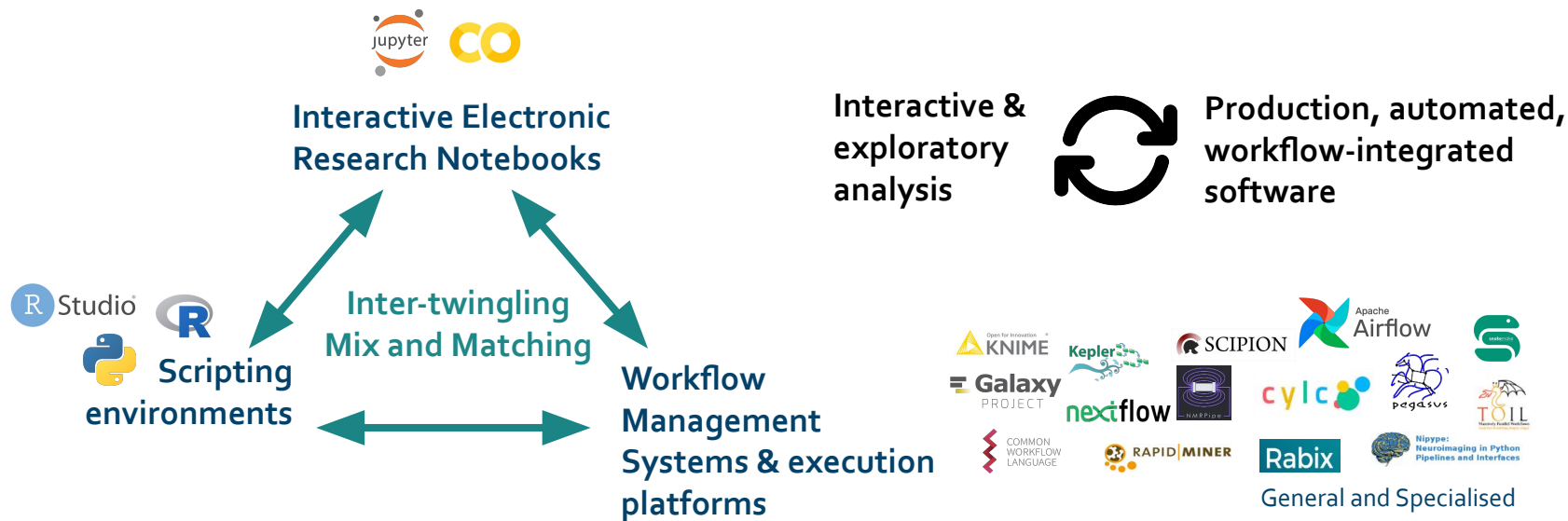
Collective working

- Building up a pool of know-how
- Variant-based development - reuse, remix, repurpose



Workflow System Landscape - what do you use/need?

Different systems have their different strengths



Frameworks to web based analysis platforms, hybrid cloud deployment



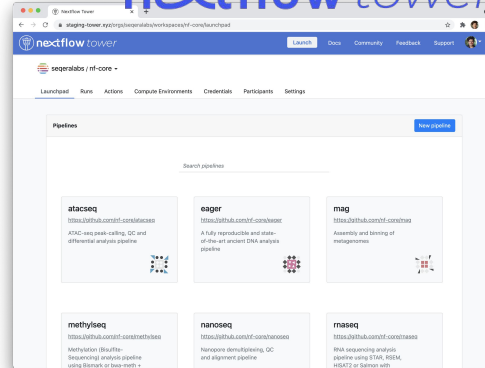
<https://snakemake.github.io/>

```
rule sort_and_annotate:
    input:
        a="path/to/{dataset}.txt",
        b="path/to/annotation.txt"
    output:
        pipe ("{dataset}.sorted.txt")
    shell:
        "paste <(sort {input.a}) {input.b} > {output}"
```

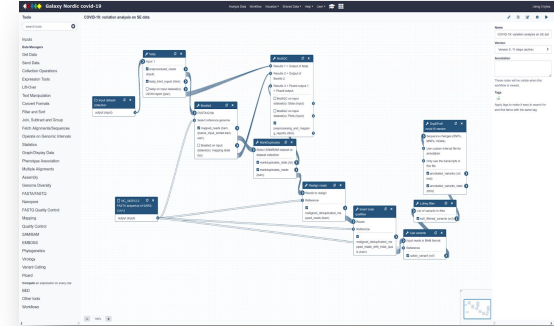
Annotations for the code block:
- 'name input and output files' points to the input and output lines.
- 'stream output' points to the pipe command in the shell line.

Workflows are rules:

Graph of jobs for automatic parallelisation,
DIY package & containerisation
installation, auto-documentation
Open framework, DSL and execution
engine using file systems and data outside



DSL Turing complete language,
open and platform



Self-contained online VRE. Users build
and reuse workflows around publicly
available or user-uploaded data and
pre-wrapped, pre-installed tools.

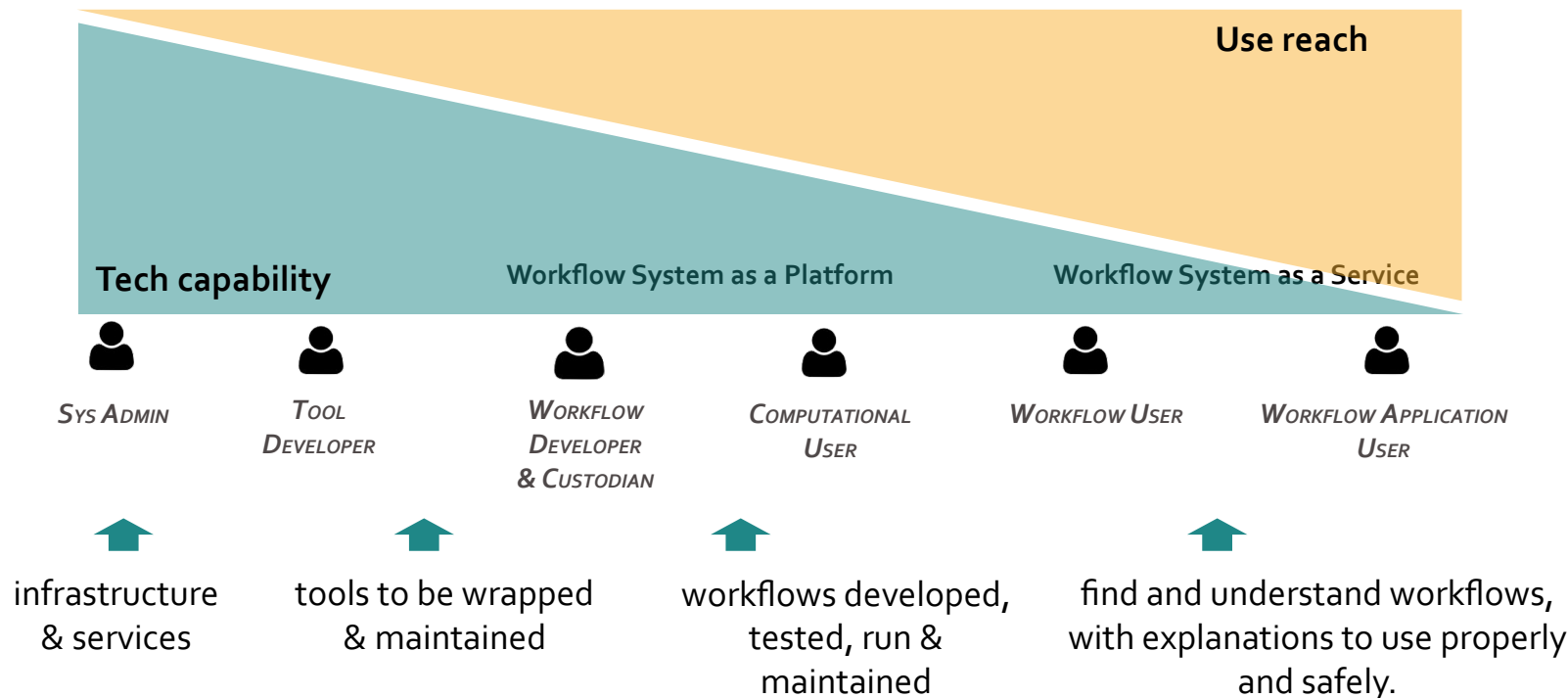
Communities tend to cluster round a few systems.

**Take up of a WfMS typically depends on the “plugged-in” availability of data type
specific codes, skills level of the workflow developers, and popularity.**



Workflow stakeholders - who are you?

Are your workflows one-offs or production ?



Questions



What workflow system do you use or need?

What kind of workflow user are you?





Now you know what a workflow is!

Mentimeter information gathering

<https://www.menti.com/87ouhg158g> or

Go to www.menti.com and use the code **4885 3343**



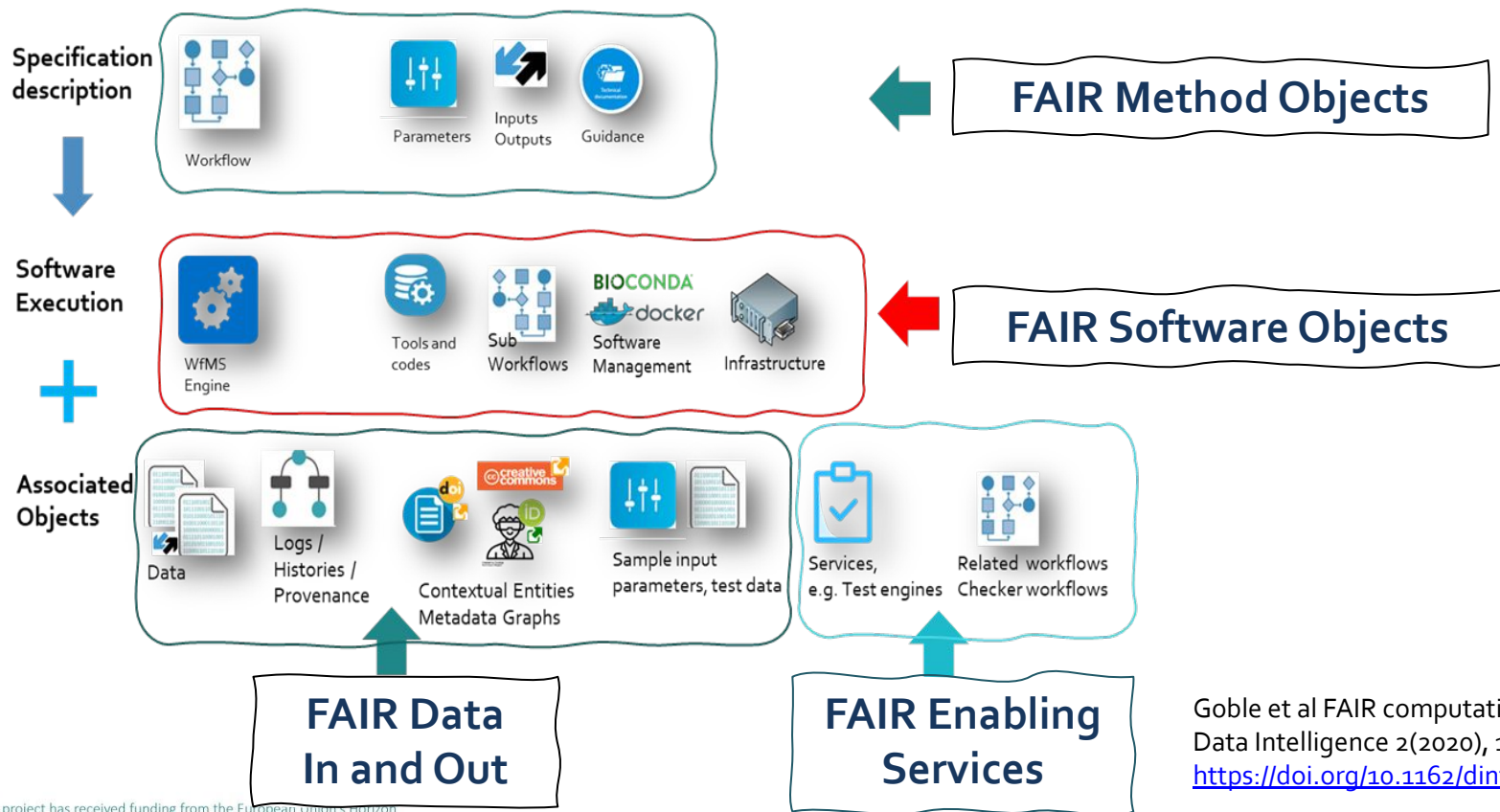


Making my Workflows FAIR



This project has received funding from the European Union's Horizon
2020 research and innovation programme under grant agreement No 824087

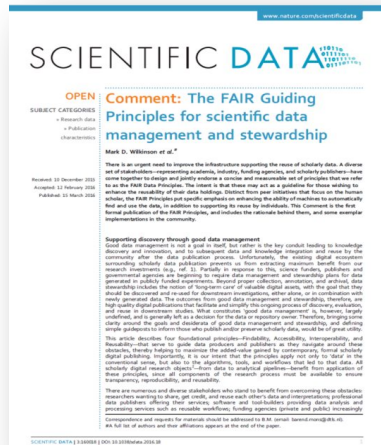
FAIR Principles applied to Workflows



Goble et al FAIR computational workflows.
Data Intelligence 2(2020), 108–121.
https://doi.org/10.1162/dint_a_00033



Workflows should produce FAIR Data Products



F, I, R Metadata generated and exposed for data products?

F1 Workflow consumes, propagates, generates FAIR **identifiers** for data?

R1 license data outputs?

R1.1 community data formats?

R1.1 usage restrictions on the reference data needed?

R1.2 parameters validated?

A1 How does the workflow **access** FAIR data?

R1.2 fully track and report data **provenance** through the workflow?



Workflow FAIR Data by Design

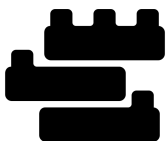


Assisted by WfMS, Challenge of diverse API & AAI landscape, formats and packaging



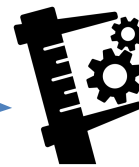
Created by monnik
from Noun Project

**Design for
FAIR Data
and Reuse**



Created by Vectors Point
from Noun Project

**Canonical
workflows**



Created by Bold Yellow
from Noun Project

**Best Practice
Golden
Examples**



Created by The Icon Z
from Noun Project

**Reviewing
Curation
Certification
Governance**

Workflow consumes, propagates, generates
FAIR **identifiers** for data?

license data outputs?

community **data formats**?

usage restrictions on the
reference data needed?

How does the
workflow **access**
FAIR data?

parameters validated?

**fully track and report data
provenance** through the workflow?

Metadata generated for data products



Things to consider when making workflows to make sure the data their produce adheres to the FAIR principles



Is the input data publicly accessible, and if so where ?

Will the output data be publicly accessible, and if so where ?

Will the data be licensed ?

Is metadata generated for the data products?

- Do you have to collect the intermediate data products too?

Is data traceable?

- combination of DIY provenance in the workflow + the WfMS support



Workflows should be FAIR objects and FAIR software



FAIR for Research Software (FAIR4RS) Working Group

[Katz, et al PATTERNS 2, 2021](#)

Patterns



Opinion

Taking a fresh look at FAIR for research software

Daniel S. Katz,^{1,*} Morane Gruenpeter,² and Tom Honeyman³

¹University of Illinois, Urbana, IL, USA

²Alma, Software Heritage, Paris, France

³Australian Research Data Commons, Melbourne, VIC 3145, Australia

*Correspondence: d.katz@leeds.org

<https://doi.org/10.1016/j.patter.2021.100222>

Software is increasingly essential in most research, and during research. To make this research software (FAIR), we need to define exactly what FAIR means: is a living and complex object for which it is impossible

Data Science 3 (2020) 37–59

DOI:10.22334/DS-190026

IOS Press

[Lamprecht et al., 2020](#)

Towards FAIR principles for research software

FAIR Computational Workflows

[Goble et al., 2020](#)

Carole Goble^{1,*}, Sarah Cohen-Boulakia², Stian Soiland-Reyes^{3,4}, Daniel Garijo⁵, Yolanda Gil⁶, Michael R. Crusoe⁷, Kristian Peters⁸ & Daniel Schober⁹

¹Department of Computer Science, The University of Manchester, Oxford Road, Manchester M13 9PL, UK

²Laboratoire de Recherche en Informatique, CNRS, Université Paris-Saclay, Batiment 650, Université Paris-Sud, 91405 ORSAY Cedex, France

³Information Sciences Institute, University of Southern California, Marina Del Rey CA 90292, USA

⁴Common Workflow Language project, Software Freedom Conservancy, Inc. 137 Montague St STE 380, NY 11201-3548, USA

⁵Leibniz Institute of Plant Biochemistry (IPB Halle), Department of Biochemistry of Plant Interactions, Weinberg 3, 06120 Halle (Saale), Germany

Kuzak^{6,d}, Carlos Martinez⁷,
Minguez Del Angel⁸,
Martinez⁹, Peter McQuilton¹,
Opoulos², Josep L.L. Gelpi^{9,e},
Pella-Gutierrez^{7,f,g}

F1

register workflows with assigned PID + metadata in a searchable resource.

A1

metadata & workflow retrievable by PID using a standardized communication protocol; metadata retrieval even if the workflow is no longer available.

I1

workflows should read, write or exchange data using domain-relevant community standards

R1

The workflow is **usable** (it can be executed) & **reusable** (it can be understood, modified, built upon, or incorporated into other workflows).

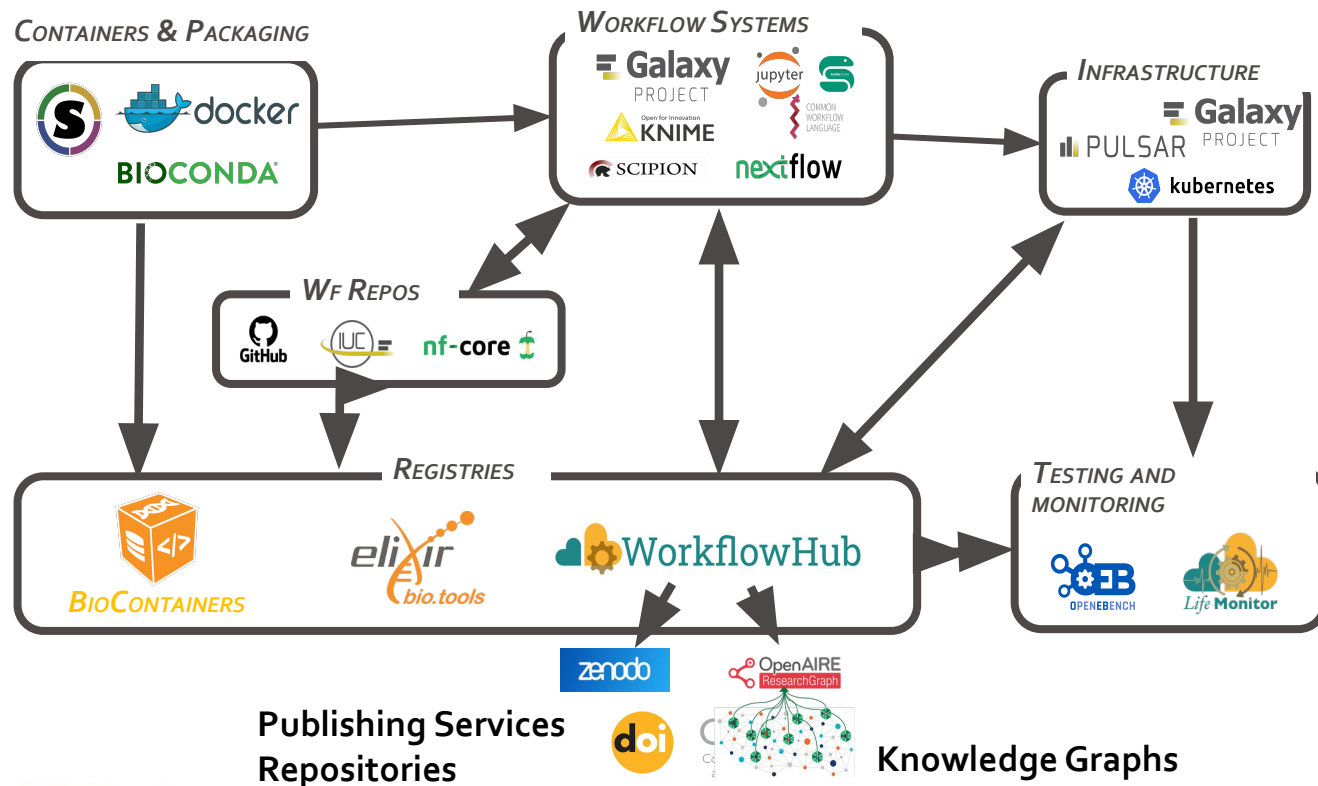
R1.2

(Meta)data, workflows and tools are associated with **detailed provenance – data lineage, workflow lineage & workflow logs**



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824087

We need Workflow Services to be FAIR and we need FAIR Enabling Services



Honour legacy & diversity by supporting native repositories & platforms

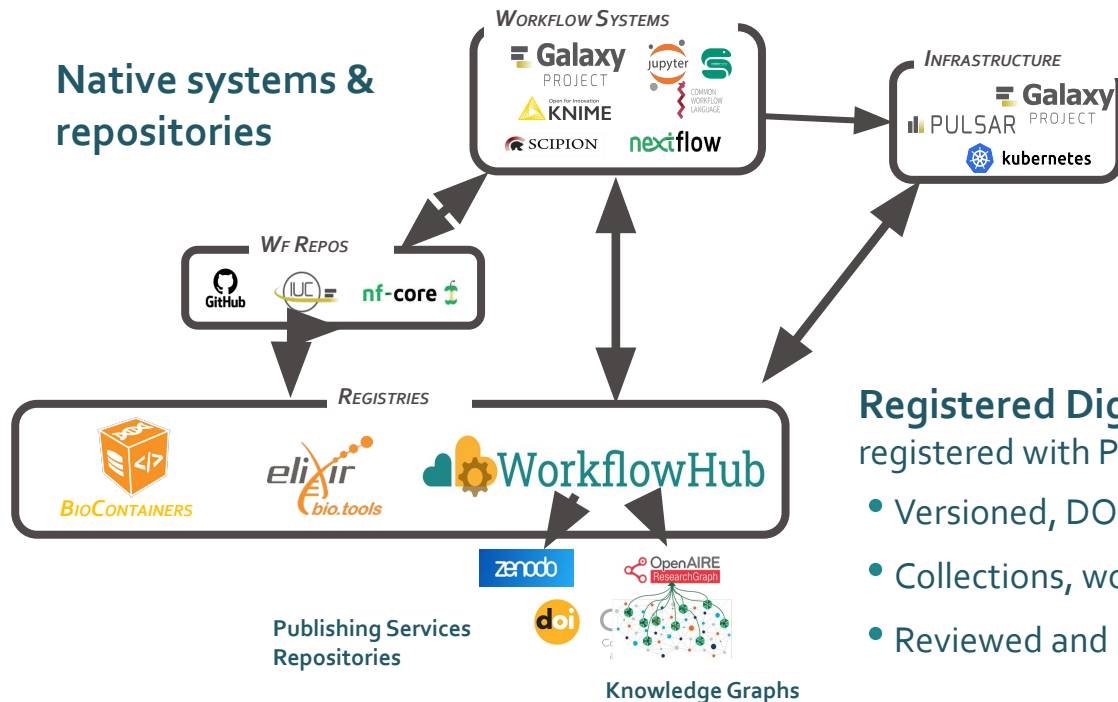
On-board systems and services:

- lifting and sharing common ids & metadata
- adopting common APIs

Enable FAIR workflow design



Workflows are Findable & Accessible



Access, Licence & Execute accessed from the PID

- Download
- Execution API (GA4GH)
- Metadata accessible even if workflow un-runnable

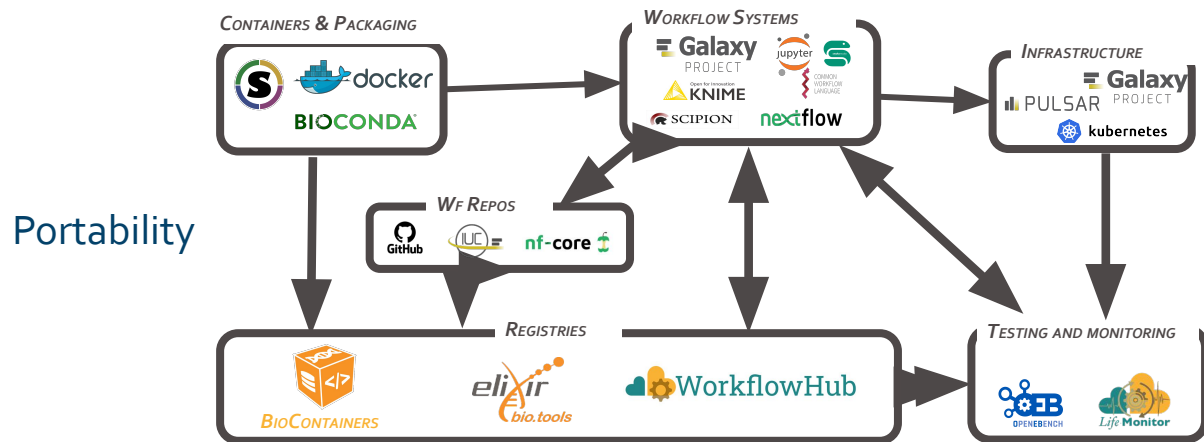
Registered Digital Objects of Scholarship

registered with PID + metadata in a searchable resource

- Versioned, DOI/PID assignment <https://workflowhub.org>
- Collections, workflow libraries
- Reviewed and Published

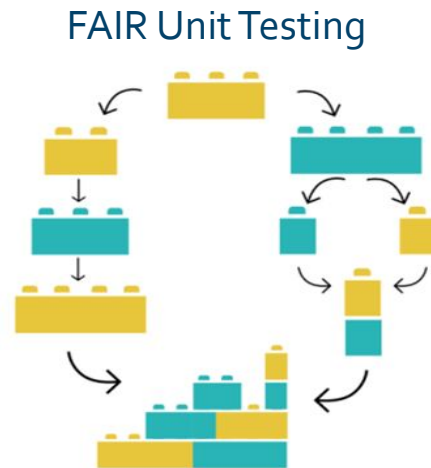


Workflows are Interoperable, Reusable and Usable



Workflow and tool collections & libraries to share, reuse and remix .

<https://workflowhub.org>



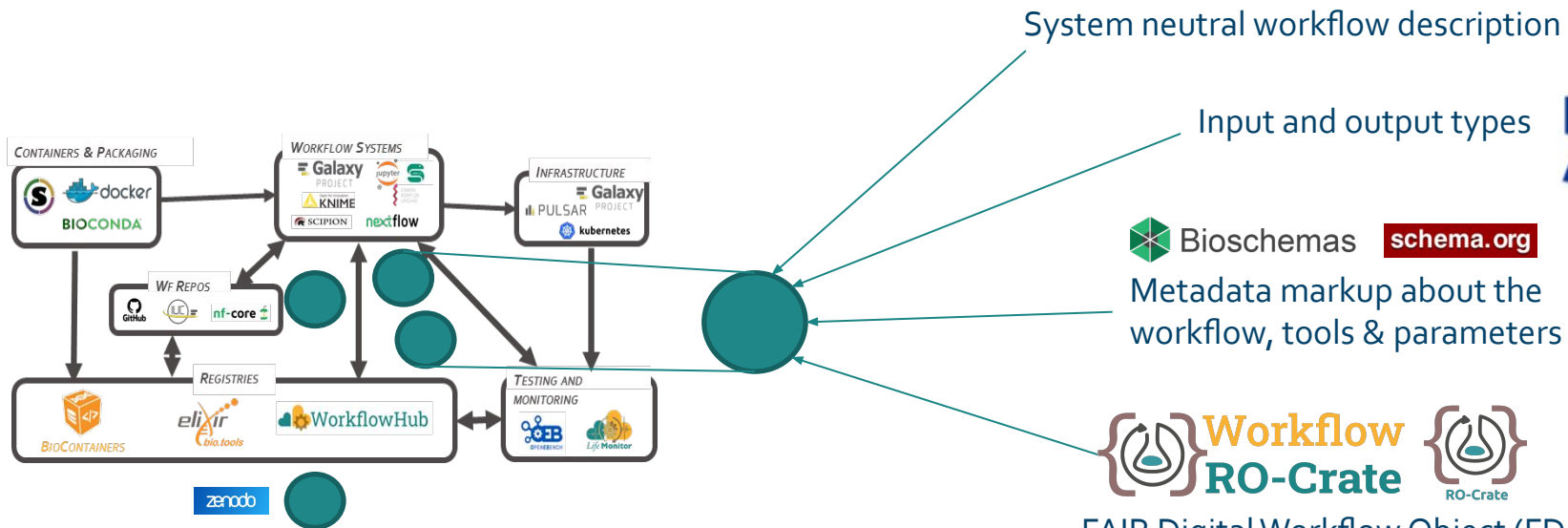
Workflow benchmarking, monitoring, testing

<https://lifemonitor.eu/>

<https://openebench.bsc.es/dashboard>



The FAIR principles are all about human and machine actionable **metadata circulating between the services** and **exported outside**



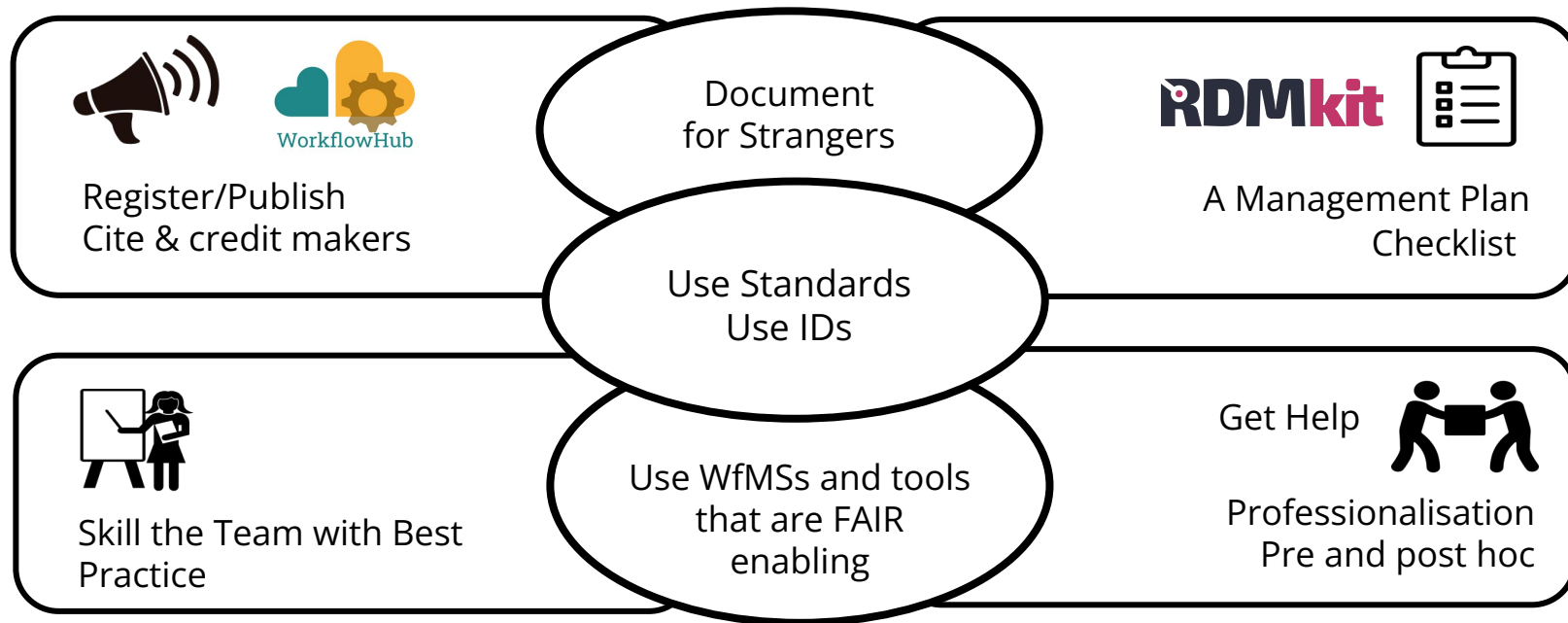
We hide most of this!





What can you do to be FAIR?

We can help each other





Registering a Workflow in WorkflowHub Using benchmarking and testing tools

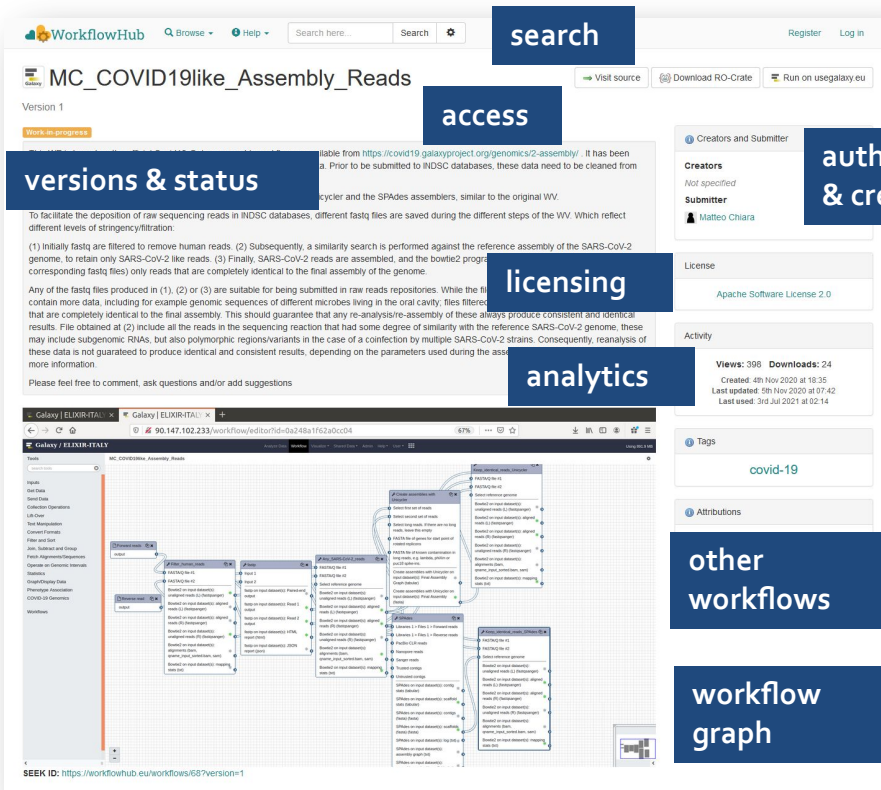




Findable & Accessible

*register workflows with
assigned PID + metadata in a
searchable resource*

DOIs on workflows
Publishable and citable objects
curated collections



Faceted Search

Rich metadata



Findable and Accessible



Curated Collections
Associated objects
Teams, Credit

Visit source

View on GitHub

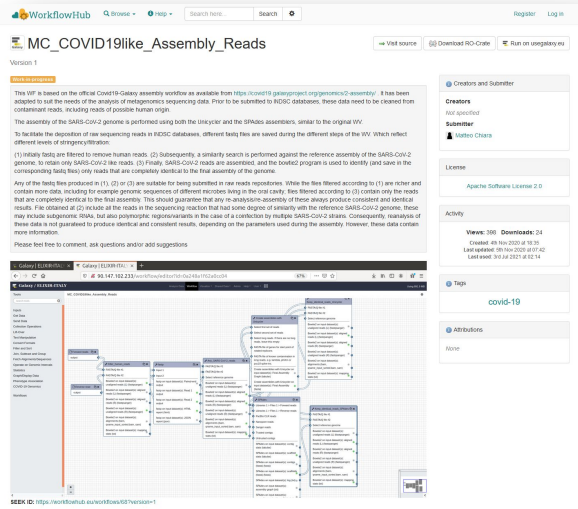
metadata & workflow retrievable
by PID using a standardized
communication protocol



Global Alliance
for Genomics & Health
Collaborate. Innovate. Accelerate.



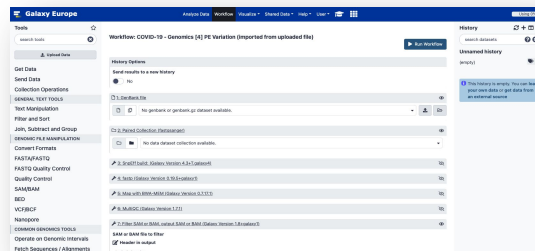
Created by Robert Schumann
Van Nieuwen Project



Download RO-Crate

Run on usegalaxy.eu

TRS



register workflows with assigned PID +
metadata in a searchable resource



teams

43 Teams found

🔍 Title (A-Z) ▼

← Previous 1 2 3 4 5 6 7 Next →

Default Condensed Table

Australian BioCommons



No description specified

Space: Australian BioCommons
Public web page: <https://www.biocommons.org.au/>

Organisms: Not specified

Ay Lab



Dr Ferhat Ay and his lab are currently located at the La Jolla Institute of Immunology. Our lab focuses on the study of the 3D genome including the development of statistical tools to better interrogate functions and associations between the 3D genome and other biological factors.

Space: Independent Teams
Public web page: <https://www.lji.org/labs/ay/>

Organisms: Not specified

BioBB Building Blocks



The BioExcel Building Blocks (biobb) software library is a collection of Python wrappers on top of popular biomolecular simulation tools. This library offers a layer of interoperability between the wrapped tools, which make them compatible and prepared to be directly interconnected to build complex biomolecular workflows. The building blocks can be used in many different workflow systems, including Galaxy, CWL, Jupyter Notebook and PYCOMPS – notably their ...

Space: BioExcel
Public web page: <https://immbb.irbbarcelona.org/biobb/>

Organisms: SARS-CoV-2

Connor Lab



Nextflow pipelines for running the ARTIC network's fieldbioinformatics tools (<https://github.com/artic-network/fieldbioinformatics>), with a focus on ncov2019

Space: COVID-19 Biohackathon
Public web page: <https://github.com/connor-lab/ncov2019-artic-nf>

Organisms: Homo sapiens, SARS-CoV-2

COVID-19 PubSeq: Public SARS-CoV-2 Sequence Resource



No description specified

Space: Independent Teams
Public web page: <http://covid-19.genenetwork.org/>

Organisms: Not specified

CWL workflow SARS-CoV-2



CWL workflows related to virus genomics with focus on SARS-CoV-2

Space: COVID-19 Biohackathon
Public web page: <https://github.com/tfmoreau/cwl-workflow-SARS-CoV-2>

Organisms: Homo sapiens, SARS-CoV-2

🔗 View on GitHub

📦 Download RO-Crate

📄 Creators and Submitter

Creators

👤 Adam Hospital, 👤 Genis Bayarri

Submitter

👤 Genis Bayarri

Discussion Channels

BioExcel Workflows

Tutorial

Documentation

Launch on MyBinder

Citation

📄 Copy

Bayarri, G., & Hospital, A. (2021). *AMBER Constant pH MD setup tutorial using BioExcel Building Blocks (biobb) (jupyter notebook)*. WorkflowHub. <https://doi.org/10.48546/WORKFLOWHUB.WORKFLOW.132.1>

American Psychological Association 7th ▼

License

Apache Software License 2.0

Organise workflows Teams and People



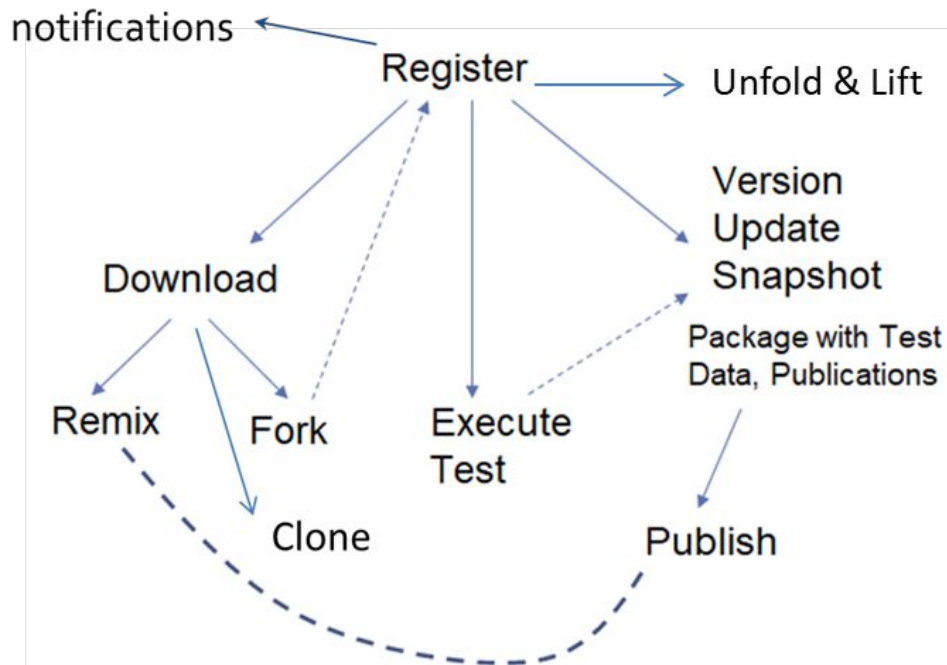
channels



Citation
Credit



Workflow lifecycle support



Curation

Manual (e.g. collections)

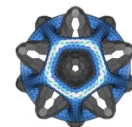
Automated (e.g. GitHub)

By stealth (e.g. from WfMS)

By services (e.g. monitoring & testing)

Credit for curators?

Work-in-progress




The Journal of Open
Source Software



What can I register in WorkflowHub?








- **Workflows!**
 - Including scripts, Jupyter notebooks etc.
- **Presentations**
- **Publications**
 - Import with DOI or pubmed ID
- **Data sets**
 - Register example/test data and associate with a workflow
- **Documents**
- **Curated collections** of all the above
 - Gather related workflows etc. together

 **BioExcel Building Blocks (BioBB)**
Protein MD Setup Tutorials

Selection of [BioExcel Building Blocks \(BioBB\)](#) [Workflows](#), across three workflow languages (jupyter notebooks, BioBB tool descriptors for each of these systems).

SEEK ID: <https://workflowhub.eu/collections/3>

Items	
	Protein MD Setup tutorial using BioExcel Building Blocks (biobb) Jupyter Notebook Workflow - Added 3 months ago
	Protein MD Setup tutorial using BioExcel Building Blocks (biobb) CWL Workflow - Added 3 months ago
	Protein MD Setup tutorial using BioExcel Building Blocks (biobb) Galaxy Workflow - Added 3 months ago
	Protein MD Setup HPC tutorial using BioExcel Building Blocks (biobb) in PyCOMPSs PyCOMPSs workflow for HPC - Added 3 months ago
	Protein MD Setup tutorial using BioExcel Building Blocks (biobb) in KNIME Experimental KNIME workflow - Added 3 months ago

<https://workflowhub.eu/collections/3>



How can I register a workflow / notebook / script in WorkflowHub?

- Multiple ways to register
 - Upload files from your computer
 - Link to existing files on the web
 - Import **git** repositories
- Flexible type system
 - Numerous **workflow types** supported
 - Metadata extraction or Abstract CWL
 - Diagram generation or upload
 - Users can add additional types if needed
 - Scripts and Notebooks welcome!

New Workflow

Currently there are two ways to register a workflow:

- **Simple** - as a file with a picture and a CWL description if you have one
- **Advanced** - as a *Workflow RO-Crate* (Note: this format is under development, click here to find out more)

Currently supported workflow types are: Common Workflow Language, Galaxy, Janis, Jupyter, KNIME, Nextflow, PyCOMPSS, Scipion, Shell Script, Snakemake.

Simple

Advanced (Workflow RO-Crate)

Workflow *

The main executable workflow.

Local file

Remote URL

Browse...

No file selected.

Workflow Type *  New workflow type

Other

The type of the above workflow.

Abstract CWL

(Optional) The abstract CWL that describes the workflow.

Local file

Remote URL

Browse...

No file selected.

Diagram

(Optional) A diagram that illustrates the main workflow.

Local file

Remote URL

Browse...

No file selected.

Register

or

Cancel

Core Workflow Types

Title	Identifier
Snakemake	https://doi.org/10.1093/bioinformatics/btq151
Nextflow	https://www.nextflow.io/
KNIME	https://www.knime.com/
Galaxy	https://galaxyproject.org/
Common Workflow Language (CWL)	https://w3id.org/cwl/

User-added Workflow Types

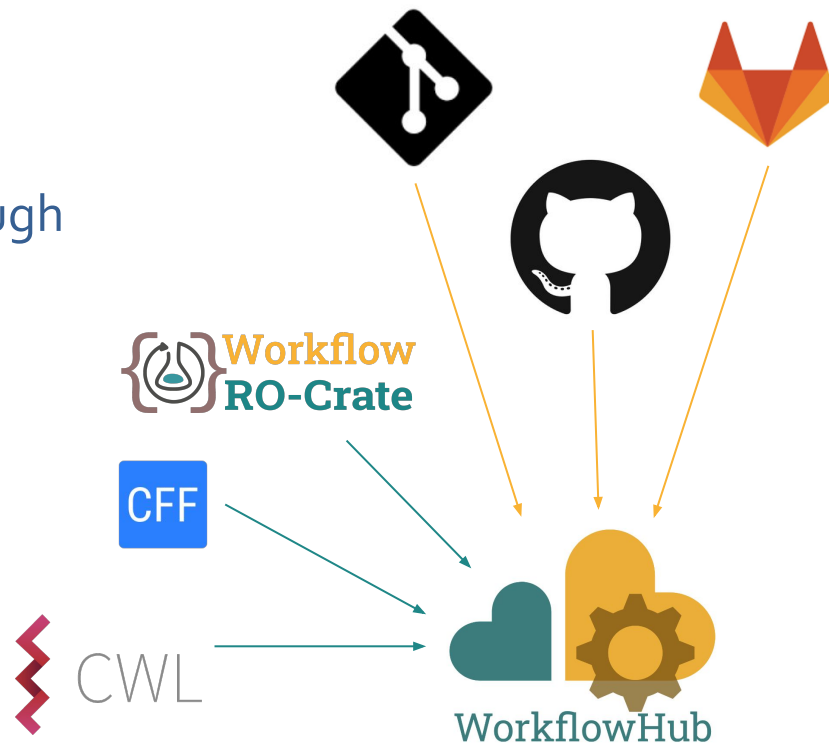
Title	Identifier	URL
Jupyter	https://jupyter.org/	https://jupyter.org/
Shell Script	Not specified	Not specified
PyCOMPSS	https://pypl.org/project/pycompss/	https://www.bsos.umd.edu/~bcooper/pycompss/
Scipion	Not specified	http://scipion.es/
Janis	Not specified	https://janis.readthedocs.io/



WorkflowHub & GitHub(/lab) Coming Soon



- “Pull” workflows directly from Git repositories
- “Push” workflows automatically through GitHub actions etc.
- Automatic parsing of metadata from repositories:
 - Workflow files
 - LICENSE
 - citation.cff
 - ro-crate-metadata.json



What metadata do I need to register a workflow?



Workflow Type *  New workflow type

PyCOMPSs

Title *

Protein MD Setup HPC tutorial using BioExcel Building Blocks (biobb) in PyCOMPSs

Minimal required metadata, extend incrementally

Description

B I H       

This PyCOMPSs workflow tutorial aims to illustrate the process of setting up a simulation system containing a protein, step by step, using the BioExcel Building Blocks library (biobb) in PyCOMPSs for execution on HPC. Three variants of the MD Setup workflows are included, supporting a list of structures, a list of mutations, or a cumulative set of mutations.



Source

https://github.com/bioexcel/biobb_hpc_workflows

If this workflow came from an external repository (i.e. GitHub), you can include its original URL here.

Maturity

Stable


This field is used to indicate to users what level of stability they can expect from the workflow.

 License 

License

Apache Software License 2.0

For more information on this license, please visit <https://opensource.org/licenses/Apache-2.0>

 Creators 

If this Workflow was created, by or together with, other members, please list their names below.

The creators you specify will have permissions to 'view and edit summary and contents'.

So far you have specified the following creators:

Adam Hospital (BioBB Building Blocks) ✖

Pau Andrio (BioBB Building Blocks) ✖

Please type creators into the box below - suggestions will be displayed as you type.

Genis Bayarr

If anyone not registered with WorkflowHub assisted creating this Workflow, you can specify them below.

What *extra* metadata could I add for a workflow?



Discussion Channels ▲

You can add one or more URL's to point to discussion channels related to this Workflow:

<https://ask.bioexcel.eu/c/bioexcel-building>

ask.bioexcel.eu

Remove

<http://mmb.irbbarcelona.org/biobb/workflc>

[BioExcel workflc](#)

Remove

<https://biobb-wf-md-setup.readthedocs.io/>

[Documentation](#)

Remove

https://github.com/bioexcel/biobb_hpc_wr

[GitHub issues](#)

Remove

[+ Add new link](#)

Tags ▲

You are able to edit all tags, including both the tags you have added and tags added by other people. Any new tags you add will be added as your own tags. Tags you remove, even those added by other people, will be completely removed from this Workflow. Tags should be separated by a comma, and known tags will appear in the drop-down box as you type.

[molecular dynamics](#) [x](#) [GROMACS](#) [x](#) [BioBB](#) [x](#)

[View suggestions](#) ▼

Attributions ▲

If this Workflow is based on any existing Workflows, please list them below

So far you have specified the following attributions:

Workflow: Protein MD Setup tutorial using BioExcel Building Blocks (biobb) (Douglas Lowe) ✕

Please type titles of Workflows into the box below - suggestions will be displayed as you type.
Select resources that you want to attribute to.

Publications ▲

The following Publications are associated with this workflow:

BioExcel Building Blocks, a software library for interoperable biomolecular simulation workflows ✕

Select Publication...



☐ Associate Publications from other projects?





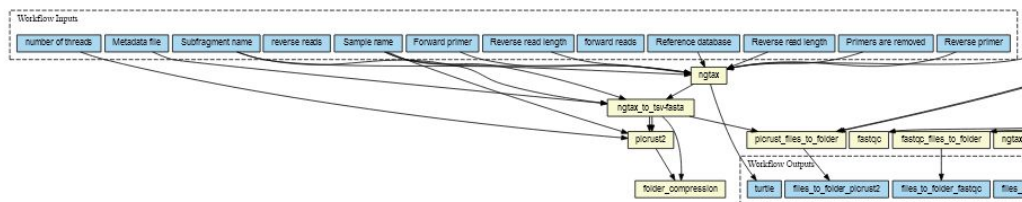
Quality assessment, amplicon classification and functional prediction

Version 2 ▾

Workflow Type: Common Workflow Language

Workflow for quality assessment of paired reads and classification using NGTax 2.0 and functional annotation using picrust2. In addition their respective subfolders for easier data management in a later stage. Steps:

- FastQC (read quality control) - NGTax 2.0 - Picrust 2 - Export module for ngtax

SEEK ID: <https://workflowhub.eu/workflows/154?version=2>DOI: [10.48546/workflowhub.workflow.154.2](https://doi.org/10.48546/workflowhub.workflow.154.2)

Inputs

ID	Name	Description
forward_reads	forward reads	forward sequence file locally
reverse_reads	reverse reads	reverse sequence file locally

<https://doi.org/10.48546/workflowhub.workflow.154.2>

Inputs

ID	Name	Description	Type
forward_reads	forward reads	forward sequence file locally	File
reverse_reads	reverse reads	reverse sequence file locally	File?
forward_primer	Forward primer	Forward primer used	string
reverse_primer	Reverse primer	Reverse primer used	string?
reference_db	Reference database	Reference database used in FASTA format	string?
rev_read_len	Reverse read length	Read length of the reverse read	int?
for_read_len	Reverse read length	Read length of the reverse read	int
sample	Sample name	Name of the sample being analysed	string
fragment	Subfragment name	Subfragment that is being analysed (e.g. V1-V3 or V5-region)	string
primersRemoved	Primers are removed	Whether the primers are removed or not from the input files	boolean?
threads	number of threads	number of threads to use for computational processes	int?
metadata	Metadata file	UNLOCK assay metadata file	File?

Steps

ID	Name	Description
fastqc		
reads_to_folder		
ngtax		
ngtax_to_tsv-fastq		
picrust2		
folder_compression		
fastqc_files_to_folder		
ngtax_files_to_folder		
picrust_files_to_folder		
phyloseq_files_to_folder		

Outputs

ID	Name	Description	Type
turtle	n/a	Used for other workflows	File
files_to_folder_fastqc	n/a	n/a	Directory
files_to_folder_ngtax	n/a	n/a	Directory
files_to_folder_picrust2	n/a	n/a	Directory
files_to_folder_phyloseq	n/a	n/a	Directory

What WorkflowHub features can be useful for my project and workflows?



- Make your workflow **discoverable**
- **Catalogue** of workflows from a project/group/organization
- “Sharing” **permissions** let you decide who can access your workflow
 - Share first with collaborators, publish when ready
- Indicate your workflow **status** as a “Work in progress” or “Stable”
- Register new **versions** as your workflow evolves
- Generate a **DOI** for your workflow and make it citable
- **Relate** workflows to ongoing and published **research**
- RO-Crate and Bioschemas generated with **FAIR metadata**



Questions to ask yourself



Will you have some workflows *ready to register* during the hackathon?

Does anything prevent your workflow being published and made **open**?

Will multiple people be **collaborating** openly on the workflow?

Do you know the names and affiliations of the workflow **authors**?

Have you considered the **license** for your workflow ?

<https://www.software.ac.uk/resources/guides/choosing-open-source-licence>



How can I test my workflows?



- Functional testing
 - Does the workflow run with no errors?
 - Does it produce the expected outputs?
- Test cases
 - Small, simple inputs → easy to make assertions on expected outputs
 - Tests should run relatively fast and not involve randomness (e.g., set seed)
- Dedicated frameworks simplify setting up test cases
 - [Planemo](#) (Galaxy and CWL)
 - [Pytest-workflow](#) (engine-agnostic)
- CI services automate test execution and provide isolated environments
 - GitHub Actions, Jenkins, Travis CI, ...
- The [LifeMonitor](#) team is working with [Galaxy IWC](#)
 - Integration of workflow best practices with WorkflowHub and LifeMonitor
 - Testbed for more general workflow maintenance support framework





Incorporating my tool and making it FAIR Workflow ready



November 1, 2021

Working paper

Open Access

10 Simple Rules for making a software tool workflow-ready

 Paul Brack;  Peter Crowther;  Stian Soiland-Reyes;  Stuart Owen;  Douglas Lowe;  Alan R Williams;  Quentin Groom;  Mathias Dillen;  Frederik Coppens;  Björn Grüning;  Ignacio Eguinoa;  Philip Ewels;  Carole Goble

Workflows have become a core part of computational scientific analysis in recent years. Automated computational workflows multiply the power of researchers, potentially turning “hand-cranked” data processing by informaticians into robust factories for complex research output.

However, in order for a piece of software to be usable as a workflow-ready tool, it may require alteration from its likely origin as a standalone tool. Research software is often created in response to the need to answer a research question with the minimum expenditure of time and money in resource-constrained projects. The level of quality might range from “it works on my computer” to mature and robust projects with support across multiple operating systems.

Despite significant increase in uptake of workflow tools, there is little specific guidance for writing software intended to slot in as a tool within a workflow; or on converting an existing standalone research-quality software tool into a reusable, composable, well-behaved citizen within a larger workflow.

In this paper we present 10 simple rules for how a software tool can be prepared for workflow use.

Accepted at PLOS Computational Biology (PCOMPBIOL-D-21-01704)

Brack, et al (2021):

10 Simple Rules for making a software tool workflow-ready.

PLOS Computational Biology (accepted)

<https://doi.org/10.5281/zenodo.5636487>



The rules



Rule 1: Make sure a **workflow engine** can talk to your software easily

Rule 2: Make your tool simple to **install**

Rule 3: **Document** your tool

Rule 4: **Make** your tool **maintainable**

Rule 5: Follow the principle of **least surprise**

Rule 6: Make your tool **parallelizable**

Rule 7: Make your workflow tool a **good citizen**

Rule 8: Make output **reproducible**

Rule 9: Carefully consider human **interaction**

Rule 10: A software tool should just **do one thing**

<https://doi.org/10.5281/zenodo.5636487>



WfMS and tool interaction



Rule 1: Make sure a **workflow engine** can talk to your software easily

Rule 2: Make your tool **simple to install**

Rule 8: Make output **reproducible**

Rule 9: Carefully consider **human interaction**

Tool *configurable* at runtime

Software and library *dependencies* should be explicit

Use a *package manager*

Does your software tool make it simple for researchers to include *all of the data*, methodology and software tooling to allow another researcher to *recreate* research findings?

Wrapping a tool for a workflow can end up hiding *visualizations* and *interactions* with users.

Human-in-the-loop sacrifices automation and reproducibility

User inputs must be reported as *parameters* or choices, so this is made accessible as *provenance* and potentially automatable later.

<https://doi.org/10.5281/zenodo.5636487>



WfMS composability



Rule 1: Make sure a workflow engine can **talk to your software** easily

Rule 5: Follow the **principle of least surprise**

Rule 10: A software tool should just **do one thing**

All **input** and **output** data is computer-addressable

Input and output file(s) or directory(ies) always be specified as *arguments at runtime*.

Inputs to and outputs clearly and explicitly *named*, tool's behaviour match documentation

Use *standard streams* in an accepted manner

Where software performs several different discrete tasks, *wrap* the executable in several different wrappers and implement different endpoints as different workflow tools.



Workflow execution



Rule 6: Make your tool **parallelizable**

Rule 7: Make your workflow tool a **good citizen**

Do multiple copies of the running software *overwrite* one another's runtime file structures?

Tool accesses *external APIs* within access limitations.

Design software likely to be run on *shared infrastructure* than on a single user's computer.

Reasonably *performant* and should not reserve more *system resources* than needed.

Disk usage post-run and clear up any *temporary* files

<https://doi.org/10.5281/zenodo.5636487>



Maintainability & Reliability



Rule 3: **Document** your tool

Rule 4: Make your tool **maintainable**

Interface-level *documentation*

Provide *code snippets*

Source control software (e.g. git)

Version management

Source code *archive*, **installation** package

Unit and Integration **tests**

<https://doi.org/10.5281/zenodo.5636487>



Acknowledgements



Join the WorkflowHub Club

<https://about.workflowhub.eu/community/>

WorkflowHub <https://workflowhub.eu/>
EOSC-Life <https://www.eosc-life.eu/>
ELIXIR <http://elixir-europe.org>
RO-Crate <https://www.researchobject.org/ro-crate/>
Galaxy Europe <https://galaxyproject.eu/>
Bioschemas <https://bioschemas.org>
CWL <https://www.commonwl.org/>
WorkflowsRI <https://workflowsri.org/>
Dockstore <https://dockstore.org/>
LifeMonitor <https://lifemonitor.eu/>
BY-COVID <https://by-covid.org/>





Questions?



This project has received funding from the European Union's Horizon
2020 research and innovation programme under grant agreement No 824087