FAIR Computational Workflows

DOI:
10.5281/zenodo.3268653
10.1162/dint_a_00033

Document Version
Submitted manuscript

Submitted to Data Intelligence special issue: FAIR best practices 2019.
This preprint: https://doi.org/10.5281/zenodo.2642531

FAIR Computational Workflows

Carole Goble[1], Sarah Cohen-Boulakia[2], Daniel Schober[3], Kristian Peters[3], Stian Soiland-Reyes[1,5], Daniel Garijo[4], Yolanda Gil[4], and Michael R. Crusoe[5]

1 School of Computer Science, The University of Manchester, Manchester M13 9PL UK
2 Laboratoire de Recherche en Informatique, UMR CNRS 8623, Université Paris-Sud, France
3 Leibniz Institute of Plant Biochemistry (IPB Halle), Department of Biochemistry of Plant Interactions, Weinberg 3, 06120 Halle (Saale), Germany
4 Information Science Institute, USC Viterbi, Marina Del Rey CA 90292, USA
5 Common Workflow Language project, Software Freedom Conservancy, NY, US

NB: We are developing a summary table for recommendations for FAIR workflows against the FAIR Principles.

Abstract (notes)

- Computational workflows are an increasingly important part of the research landscape, and a key tool for: instrumentation data capture, data processing pipelines, data analytics, predictive modelling and simulation suites.
- Properly designed workflows contribute to FAIR data principles [FAIR principles explained in this issue], since they provide the metadata and provenance necessary to describe their data products and they describe the involved data realms in a formalized, completely traceable way.
- Workflows are method digital objects in their own right that are FAIR too; however they are not data, they are software. The FAIR principles for data are not directly applicable and need to be adapted and extended.
- Workflows bring the FAIR principles to a new level to cater for their composite and living nature, their dependencies on their environment and their components, and their need for robust and portable execution.

Introduction

In data intensive science, e-infrastructures and software tools are heavily used to help scientists manage, analyze, and share increasing volumes of complex data [Atkinson 2017]. Data processing tasks like data cleansing, normalisation and knowledge extraction need to be automated stepwise in order to foster performance, standardisation and re-usability. Increasingly complex data computations and parameter-driven simulations need infrastructures and consistent reporting to enable systematic comparisons of alternative setups [Deelman 2017]. As a response to these needs, the practice of performing computational processes using workflows has taken hold in different domains such as the Life Sciences [Cohen-Boulakia 2017], biodiversity [Mathew 2014], astronomy [Freudling 2013], geosciences [Duffy 2012], social sciences [Turner 2015] and more generally machine learning systems such as TensorFlow.

A workflow is a precise description of a procedure – a multi-step process to coordinate multiple tasks. In computational workflows, each task represents the execution of a computational process, such as running a code, the invocation of a service, the calling of a command line tool, access to a database, submission of a job to a compute cloud, or even the execution of another workflow. Figure 1 gives an example of a real workflow represented using the Common Workflow Language[1] [Amstutz 2016].

Computational workflows promise support for automation that scales across computational infrastructures and large datasets and shield users from underlying execution complexities and inter-resource incompatibilities. From an execution perspective they are a means to handle the work of accessing an ecosystem of software and platforms, managing data, securing access, and handling heterogeneities. From a reuse and reproducibility perspective the automated methods can be packaged and ported across computational platforms easing how we can create and execute workflows in different environments and among diverse skill/expertise levels of users. From a reporting perspective they are a means to specify and document the workflow design and report the methodology: accurately recording the data inputs, parameter configurations and history of their runs and the provenance of their output data products [Cuevas-Vicenttín 2012]. The provenance of a result (i.e., why and how a given result has been obtained by an analysis) enables the understandability and comparison of multiple results, and facilitates the exchange, standardisation and reusability of results.

The rise in the use of workflows has been accompanied by the many diverse systems by which they can be implemented. At one end of the spectrum are ad-hoc scripts (Command Line, Python, Java, etc.) and interactive notebooks -which provide an intuitive interface to quickly interact with the analysis results (e.g., Jupyter[2], RStudio[3], Zeppelin[4]). At the other end are highly featured workflow management systems (WfMS) some aimed at general application s (e.g. KNIME[5]) whilst others have been adopted by specific communities with specialised features and component collections (Nipype[6] for neuro-bioimaging).

[1] http://commonwl.org/
[2] https://jupyter.org/
[3] https://www.rstudio.com/
[4] https://zeppelin.apache.org/
[5] https://www.knime.com/
[6] https://nipype.readthedocs.io/

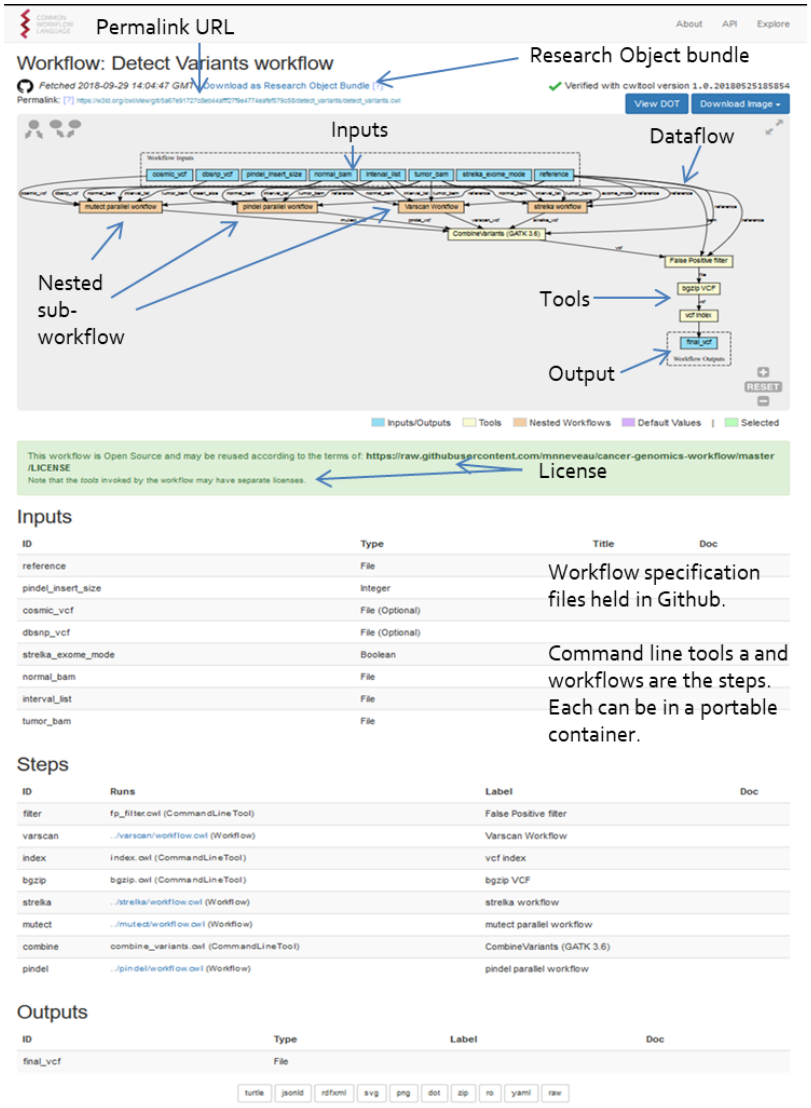

Figure 1: A workflow for detecting variants in genome sequences.
The workflow is specified in the Common Workflow Language viewed using the CWL Viewer.
Any CWL compliant execution engine should be able to execute this workflow.
https://w3id.org/cwl/view/git/5a67e91727c8eb44afff27f9e4774eafef579c58/detect_variants/detect_variants.cwl

WfMS can roughly be divided into *coarse-grained*, with a prime focus on chaining locally hosted or distributed tools (e.g. Galaxy[7], KNIME, Taverna[8]) and *fine-grained* focusing on optimising computational resources over Distributed Computing Infrastructures or HPC for applications (Pegasus[9], Spark, SnakeMake[10], Nextflow[11], Dispel4Py[12]) and cloud-based container orchestration

[7] https://galaxyproject.org/
[8] http://taverna.org.uk
[9] https://pegasus.isi.edu/
[10] https://snakemake.readthedocs.io/en/stable/
[11] https://www.nextflow.io/
[12] https://pypi.org/project/dispel4py/

(Kubernetes[13]). Many WfMS mix the two kinds [Moreno 2019]. All WfMSs aim to handle common cross-cutting concerns on behalf of the workflow execution. Concerns include: resource scalability (optimisation, concurrency and parallelisation), secure execution (of tools in their environment, monitoring and fault handling); tracking (process logging and data provenance tracking) and data handling (secure access, movement, reference management). WfMSs vary how their users interact with them, for example by APIs or command lines, or for those with more limited programming skills, a GUI for authoring the workflow specification by "drag, drop and linking". WfMS may execute over HPC or geographically distributed clusters, clouds across systems, or even from desktops. They consequently vary in their mechanisms to prepare their components to become executable steps and must manage their portability and their dependencies on the infrastructure used to run them.

Workflows are composed of modular building blocks that have been prepared with standardised interfaces to be linked together and run by a computational engine. Thus the key characteristic of computational workflows is *the separation of the workflow specification from its execution*, capturing the control flow order between components and explicitly exposing the dataflow and data dependencies between the inputs and outputs of the processing steps. This separation is fundamental to supporting workflow comprehension, design modularity, workflow comparisons and alternative execution strategies. A WfMS explicitly makes this distinction. Interactive notebooks can do so when organized appropriately by defining their dataflow in the form of interactive computational cells; i.e., input and output variables explicit in each cell, data dependencies are explicit on each cell, and the steps are executed in order. Notebooks can also be used as "meta workflows" as the steps can be scripts or command-line calls to a WfMS. Scripts tend to interleave data and computational processes, although systems such as YesWorkflow [McPhillips 2015] seek to provide users of scripting languages with the means to annotate existing scripts with special comments that reveal their hidden computational modules and dataflows.

We propose that FAIR Principles apply to workflows, and WfMSs, in two major areas:

- Properly designed workflows contribute to FAIR data principles, since they provide the metadata and provenance necessary to describe their data products and they describe the involved data realms in a formalized, completely traceable way.
- Workflows are Digital Objects in their own right, encapsulating methodological know-how that is to be found and published, accessed and cited, exchanged and combined with others, and reused as well as adapted.

These two aspects are explored of the rest of the article. References to FAIR principles [Wilkinson 2016] are given in brackets.

FAIR Data and Workflows

Key contributions of workflows to FAIR Data compliance lies in the standardisation of practices, in a world of expanding and diversifying processing tools and computational operating environments, and in the formal computer-interpretable capture of provenance data. Computational workflows are at their heart automated data processing machinery. The effectiveness of that automation is not only enhanced by FAIR data, automation changes expectations with respect to data quality, clear identification, explicitness of data organisations, structures and semantics, machine-readable

[13] https://kubernetes.io/

licenses and access permissions. FAIR data would enable a WfMS to automatically make *informed choices* from the phase of the workflow design (e.g., by suggesting tools fitting data features when several alternative tools could be considered for a given data analysis step) to the phase of workflow execution (e.g., by validating the data against a step's expected type). A WfMS thus needs to be able to access precise information on data origin, the way of accessing it, and a set of associated metadata. Domains such as the Life Sciences have developed ontologies, vocabularies and services for data interoperability (I3) and identifier resolution (F1), i.e., efforts such as the Breeding API (BrAPI)[14] to standardise the interface for exchanging data between applications and the EDAM ontology[15] to precisely specify the input and output of tools executed in a workflow (see FAIRsharing.org for examples). The formalised and finer grained annotation of data carries a cost, which is arguably why a significant amount of workflow processing is still metadata wrangling, format transformations and identifier mapping [Garijo 2014].

The workflow specification itself can be thought of as "prospective provenance", that is, as a recipe to produce a data product that exposes the extent of the effort made to make the data FAIR that could be validated against emerging FAIR indicators [Metrics paper in this issue]. Determining whether the data produced by a workflow is FAIR is not straightforward and requires concrete criteria.

The combination of FAIR data, FAIR tools (inputs, outputs, disclosure of the task, statistics) with FAIR Infrastructure [Hardisty in this issue] (metadata are available on the underlying resources for running workflows and managing results) would significantly assist in the operation of workflows. Examples include annotations on tools and libraries (e.g. Bio.tools[16], Bioconductor, CRAN, PyPI) and on software containers (e.g. Biocontainers[17]). Standardised specifications on handling data formats and executables, automated handling of tool dependencies, and versioning and explicit metadata on computational resource needs would aide harmonisation of software tools execution and efficient job scheduling and data movement throughout different FAIR e-infrastructures.

For data generation a standardised workflow specification and automated execution contributes to transparency, reproducibility, analytic validity, quality assurance and the attribution and comparison of results. If well designed workflows can automate the production of metadata descriptions of data products (F2, I2, I3, R1.3) and the deposition of data in searchable resources (F4). Identifiers, licensing and access present interesting challenges in workflow execution:

- *Identifiers* (F1, F3, A1) concerns include the propagation of identifiers through the workflow, tracking data citation [Groth in this issue] and the minting of identifiers for large numbers of intermediate results. Minids [Chard 2016] are proposed as light weight identifiers to unambiguous name, identify and reference research data products that can then optimise data exchange by reference this reducing unnecessary or insecure data movement. Workflows need to move data references through their engines not the data itself.

[14] https://brapi.docs.apiary.io/
[15] http://edamontology.org/
[16] http://bio.tools/
[17] https://biocontainers.pro/

- *Licensing* (R1.1). As workflows often combine data, data licenses need to be respected, honoured and prop, as do licenses on the software used by the workflow tasks. Combining licenses is particularly tricky and can impact on the ability to license the workflow itself or its data products.
- *Data access* (A1.1, A1.2): Single sign on to WfMS requires harmonised AAI propagation through the different tasks of the workflow which may be hosted by different service providers using different systems.

Workflows intrinsically provide precise documentation of how the data has been generated (R1.2). The detailed record of the details of every executed process together with comprehensive information about the execution environment used to derive a specific data product is retrospective provenance, either observed by the WfMS or disclosed to it by the computational task itself. A great deal of work has focused on provenance tracking of [Herschel 2017] leading to standardisation efforts such as Prov [Khan 2019]. Challenges remain: provenance standards have yet to be fully embraced by WfMS, there are shortages of provenance processing tools, and automated provenance collection can be too fine-grained and too detailed to be of service to researchers [Alper 2018]. The computational steps are themselves unFAIR. Although open source tools allow us to inspect procedures, many codes (especially those only available as run-time binaries) are black boxes or proprietary software that do not disclose the link between their inputs and outputs, breaking the provenance lineage of data. Steps in coarse-grained workflows are often wrapped applications with buried workflows and manual steps within. Data resources and tools do not always report basic metadata such as their version or licence in a standardised, machine processable way. Bioschemas[18] aims to get such metadata marked-up in resources in a lightweight way. A greater problem is unFAIR service provision, whereby the components change their interfaces without notice breaking the workflows that use them. Given their data focus, the FAIR principles are chiefly focused on the availability of metadata rather than the quality of service of the databases, tools and the e-infrastructures the data exist within.

FAIR Workflows as Digital Objects

The initial FAIR criteria were envisioned for data. As workflows are digital objects in their own right it is natural to draw an analogy with data and to try to apply the FAIR Principles to them. The majority of workflows are not yet registered in specialised repositories or are stored in software repositories indistinguishable from other software. Conventions for naming workflows still have to be devised (F1). Workflows range in the quality of their documentation, as are other software, described using proprietary or native programming languages.

Researchers have been actively exploring ways to for workflows to be FAIR. Workflow registries and repositories typically cater for specific WfMSs, such as KNIMEHub[19] for KNIME and nf-core[20] for Nextflow Life Science pipelines, to support findability and accessibility (F4), with description and metadata associated with deposited workflows (F2) and in some cases persistent, unique identifiers

[18] http://bioschemas.org/
[19] https://hub.knime.com/
[20] https://nf-co.re/

(F1). Access is typically baked into the workflow applications (A1), for example only Galaxy workflows are only available in dedicated Galaxy installations such as Workflow4Metabolomics[21]. Others such as (e.g. WINGS[22]) provide means to export workflows as Linked Data. Workflow findability in repositories has been studied [Starlinger 2012] alongside workflow similarity [Starlinger 2014] where workflows are compared based on their metadata and structure. For workflows to be accessible in the same way as data, they need to be archived and cited just as data is archived and cited using citation metadata [Smith 2016]. In the schema.org mark-up used by citation infrastructures such as Datacite, terms indicate data is derived from other data. Ideally there should also derived from terms indicating the software or service used to perform that transformation harmonised with workflow provenance.

myExperiment[23] [De Roure 2009] is an attempt at a WfMS agnostic repository, pioneering approaches for workflow sharing and publishing with licenses, crediting authors when workflow designs were reused or repurposed, and packaging workflows into collections and with other digital objects such as associated data files and publications. The work laid the foundations for workflow based Research Objects[24] [Belhajjame 2015] that allows for bundling of all the artefacts associated with an investigation or piece of research into one whole that can also be cited. The Figure 1 workflow's description files and links to executable containers and data files can be downloaded in a Research Object zip-based bundle along with citation metadata and assigned a DOI. The European Open Science Cloud for Life Sciences has started work to build a workflow registry using CWL with Research Objects federated with registries for tools (bio.tools) and containers (Biocontainers).

Several attempts have been made to standardise workflow descriptions in order to aid discoverability (F2) and enable interoperability (I1). The Interoperable Workflow Intermediate Representation [Plankensteiner 2011] was proposed as a common bridge for translating fine-grain workflows between different languages independent of the underlying distributed computing infrastructure [Terstyanszky 2014]. GA4GH Workflow Execution Service and Task Execution Schema[25], the Workflow Description Language[26] and the Common Workflow Language (CWL) are recent community efforts to describe workflows. The CWL specification aims to describe workflows and command-line tool interfaces in a way that makes them portable and scalable across a variety of software and hardware environments and runnable by other CWL-compliant engines. This last point is critical. Workflow are not data, they are *software* that are intended to be executed. Workflows as software challenge the FAIR principles by their structure, forms, versioning, executability, and reuse.

Structure. Workflows are inherently composite whose components can be workflows in a nested, fractal way. Workflows will be reused as sub-workflows. The distinction between a workflow and its components steps [Haendel 2016] is blurred. FAIR criteria can thus be applied simultaneously on multiple levels. To render a workflow findable relies on the findability of the involved tools and data

[21] https://workflow4metabolomics.org/
[22] http://www.wings-workflows.org/
[23] http://myexperiment.org/
[24] http://researchobject.org/
[25] https://www.ga4gh.org/
[26] https://software.broadinstitute.org/wdl/

types as researchers often use these as search attributes. FAIR properties on the components - metadata, licensing, author credit, access authorization and so on – propagate to the workflow level and may be incompatible. Fundamentally, how we identify, cite and credit composite software is an open question [Katz 2014].

Forms. When we speak of a FAIR workflow what do we mean? A workflow can be a CWL specification with test or exemplar data; an implementation of that design in a WfMS; an instantiation of that implementation ready to be run with input data and parameters set and computational services spun up; a run result with intermediate and final data products and provenance logs. Workflow-centric Research Objects attempt to create a metadata framework to capture and aggregate each form, but each may have different FAIR criteria.

Versioning. Software is a living artefact to be maintained, updated, and eventually deprecated. The components, the WfMS itself and the underlying computational infrastructures they run on evolve and change. Workflow evolution is a form of provenance [R1. 2] that tracks any alteration of an existing workflow resulting in another version that may produce the same or different results [Casati 1998]. Moreover, to make methodological variants workflows will be recycled and repurposed: cloned, forked, merged and dramatically changed. Workflow repositories such as nf-core embrace this software nature, building on top of collaborative development environments such as github that natively support versioning as well as testing and validation. FAIR principles have to address versioning and "fixivity" – the need to snapshot a workflow to fix its reproducible state and associate a persistent identifier.

Executability. Workflows are executable objects. To be interoperable (I1) and reusable (R1) they need to be portable, encapsulating all their runtime dependencies. Lightweight container-based virtualisation solutions to distribute software and share execution environments (e.g. CONDA, Docker, Singularity) revolutionise workflow reusability. Nevertheless, workflows are time limited objects whose active lifespan is dependent on that of their components and WfMS as much as on their scientific relevance. Consequently CWL addresses both explicit support for containerised execution and the lifting of workflow description from the WfMS or application it is embedded in so that it may be runnable in other CWL-compliant engines even when no longer executable in its native form (A2). In workflow e-infrastructures (e.g. in local workstations or in cloud environments), resource limitations need to be defined by the workflow. Implicit security aspects and stability of the workflow environment need to be covered by the infrastructure and its components and not by the workflow.

Reuse. Reusing workflows involves a continuum of situations from pure *redo* where the exact same workflow (same tools) is re-executed in the exact same environment with the exact same data and parameter settings, to *workflow replication*, where minor changes can be made usually in the workflow environment and/or parameter settings but the results remain the same, to *workflow reproducibility* where the aim of the analysis remains the same but the means (steps) or data may vary, to eventually *workflow reuse* where only part of the original workflow is considered with a possible different aim in *workflow repurposing* [Garijo 2017, Wroe 2006]. Regardless of intent, the workflow user must be confident that the expected results are generated. R1 means robust software practices [Artaza 2016, Taschuk 2017, Leprevost 2014] that entails:

- Proper testing of the computational workflow and its modules as well as the software tools that are invoked during workflow runtime;
- Validation of interoperability claims that tests workflow replication on different platforms;
- Validation of parameters to preclude workflow failure and faulty or unsafe results. The formulation of parameter must therefore be FAIR and must include documentation and explanation of their purpose and range definitions (testing of parameter ranges). The BioCompute Object specification [Alterovitz 2018] emphasises detailed representation and validation of parameters for regulatory approval of reusable computational pipelines for precision medicine.

These thoughts lead to two conclusions (i) that treating FAIR workflows as data artefacts only goes so far, and that FAIR software principles should built on best practices for software maintainability, maturity and computation reproducibility guidelines [Stodden 2016] (ii) the individual parts, forms, versions and execution environments of a workflow need to be FAIR by themselves, leading to complex interdependencies which need to be covered by FAIR metrics.

Conclusions (Notes)

- Properly designed workflows contribute to FAIR data practices, since they provide the metadata and provenance necessary to describe their data products and they describe the involved data realms in a formalized, completely traceable way. The processing and production of FAIR data is not scalable without automation there remain major challenges. The transformational power of workflows for FAIR data.
- Workflows are Digital Objects in their own right [REF Wittenburg paper in this issue], encapsulating methodological know-how that is to be found and published, accessed and cited, exchanged and combined with others, and reused as well as adapted. FAIR principles for software are different to data, and so will be their FAIR metrics/indicators, FAIR software is actually maintainable software using best software practices [Artaza 2016, Taschuk 2017] for maintainability and maturity; Reproducibility is essential
- FAIR needs to be built in to WfMS and embedded in tools and practices and that will cost

Acknowledgements (TBC)

KP is funded by the German Network for Bioinformatics Infrastructure (de.NBI) and acknowledges BMBF funding under grant number 031L0107. SS-R is funded by BioExcel2, CAG is funded by BioExcel2, ELIXIR-EXCELERATE and EOSCLife.

Peter Wittenburg, Tim Clark

References

[Cohen-Boulakia 2017] Sarah Cohen-Boulakia, Khalid Belhajjame,,Olivier Collin, Jérôme Chopard, Christine Froidevaux, Alban Gaignard, Konrad Hinsen, Pierre Larmande, Yvan Le Bras Frédéric Lemoine, Fabien Mareuil, Hervé Ménager, Christophe Pradal, Christophe Blanche **Workflows for computational reproducibility in the life sciences: Status, challenges and opportunities**, Future Generation Computer Systems 75, October 2017, 284-298, https://doi.org/10.1016/j.future.2017.01.012

[Deelman 2017] Ewa Deelman, Tom Peterka, Ilkay Altintas, Christopher D Carothers, Kerstin Kleese van Dam, Kenneth Moreland, Manish Parashar, Lavanya Ramakrishnan, Michela Taufer, Jeffrey Vetter **The future of scientific workflows**, April 2017 The International Journal of High Performance Computing Applications, https://doi.org/10.1177/1094342017704893

[Khan 2019] Farah Zaib Khan; Stian Soiland-Reyes; Richard O. Sinnott; Andrew Lonie; Carole Goble; Michael R. Crusoe, **Sharing interoperable workflow provenance: A review of best practices and their practical application in CWLProv**, GigaScience *in review,* https://doi.org/10.5281/zenodo.1966881

[Belhajjame 2015] Khalid Belhajjame, Jun Zhao, Daniel Garijo, Matthew Gamble, Kristina M. Hettne, Raúl Palma, Eleni Mina, Óscar Corcho, José Manuél Gómez-Pérez, Sean Bechhofer, Graham Klyne, Carole A. Goble: **Using a suite of ontologies for preserving workflow-centric research objects.** J. Web Sem. 32: 16-42 (2015), https://doi.org/10.1016/j.websem.2015.01.003

[De Roure 2009] David De Roure, Carole Goble, Robert Stevens, **The design and realisation of the myExperiment Virtual Research Environment for social sharing of workflows**, Future Generation Computer Systems, 25 (5) May 2009, Pages 561-567 https://doi.org/10.1016/j.future.2008.06.010

[Atkinson 2017] Malcolm Atkinson, Sandra Gesing, Johan Montagnat, Ian Taylor **Scientific workflows: Past, present and future**, (2017) https://doi.org/10.1016/j.future.2017.05.041

[Amstutz 2016] Peter Amstutz, Michael R. Crusoe, Nebojša Tijanić (editors), Brad Chapman, John Chilton, Michael Heuer, Andrey Kartashov, Dan Leehr, Hervé Ménager, Maya Nedeljkovich, Matt Scales, Stian Soiland-Reyes, Luka Stojanovic (2016): **Common Workflow Language, v1.0. Specification**, Common Workflow Language working group. https://w3id.org/cwl/v1.0/ https://doi.org/10.6084/m9.figshare.3115156.v2

[Cuevas-Vicenttín 2012] Víctor Cuevas-Vicenttín, Saumen Dey, Sven Köhler, Sean Riddle, Bertram Ludäscher, Scientific Workflows and Provenance: Introduction and Research Opportunities, Datenbank Spektrum 2012 nov; 12(3):193–203, https://doi.org/10.1007/s13222-012-0100-z

[Plankensteiner 2011] Kassian Plankensteiner, Johan Montagnat, Radu Prodan**. IWIR: A Language Enabling Portability Across Grid Workflow System**s. Workshop on Workflows in Support of Large-Scale Science(WORKS'11), Nov 2011, Seattle, United States. ACM, pp.97-106, 2011, https://doi.org/10.1145/2110497.2110509

[Terstyanszky 2014] Terstyanszky, G., Kukla, T., Kiss, T., Kacsuk, P., Balasko, A. and Farkas, Z. 2014. **Enabling Scientific Workflow Sharing through Coarse-Grained Interoperability**. Future Generation Computing Systems: The International Journal of Grid Computing and eScience. 37, pp. 46-59. https://doi.org/10.1016/j.future.2014.02.016

[Smith 2016] Smith AM, Katz DS, Niemeyer KE, FORCE11 Software Citation Working Group. 2016. **Software citation principles**. PeerJ Computer Science 2:e86 https://doi.org/10.7717/peerj-cs.86

[Casati 1998] Casati, F., Ceri, S., Pernici, B., Pozzi, G. (1998a): **Workflow Evolution**. Data and Knowledge Engineering, 24(3): 211-238 https://doi.org/10.1016/S0169-023X(97)00033-5

[Mathew 2014] Mathew C, Güntsch A, Obst M, Vicario S, Haines R, Williams A, de Jong Y, Goble C (2014**) A semi-automated workflow for biodiversity data retrieval, cleaning, and quality control.** Biodiversity Data Journal 2: e4221. https://doi.org/10.3897/BDJ.2.e4221

[Stodden 2016] Victoria Stodden, Marcia McNutt, David H. Bailey, Ewa Deelman, Yolanda Gil, Brooks Hanson, Michael A. Heroux, John P.A. Ioannidis, Michela Taufer **Enhancing reproducibility for computational methods Science** 09 Dec 2016: Vol. 354, Issue 6317, pp. 1240-1241 DOI: https://doi.org/10.1126/science.aah6168

[Alterovitz 2018] Alterovitz G, Dean D, Goble C, Crusoe MR, Soiland-Reyes S, Bell A, et al. (2018) **Enabling precision medicine via standard communication of HTS provenance, analysis, and results.** PLoS Biol 16(12): e3000099. https://doi.org/10.1371/journal.pbio.3000099

[Artaza 2016] Artaza H, Chue Hong N, Corpas M *et al.* **Top 10 metrics for life science software good practices** [version 1; peer review: 2 approved]. *F1000Research* 2016, 5(ELIXIR):2000 https://doi.org/10.12688/f1000research.9206.1

[Taschuk 2017] Morgan Taschuk, Greg Wilson, **Ten simple rules for making research software more robust,** 2017 PLoS Comp Bio https://doi.org/10.1371/journal.pcbi.1005412

[Leprevost 2014] Leprevost Fda et al On best practices in the development of bioinformatics software Front Genet. 2014 Jul 2;5:199. https://doi.org/10.3389/fgene.2014.00199

[Freudling 2013] W. Freudling, M. Romaniello, D. M. Bramich, P. Ballester, V. Forchi, C. E. García-Dabló, S. Moehler and M. J. Neeser, **Automated data reduction workflows for astronomy: The ESO Reflex environment**, November 2013, J Astronomy and Astrophysics 559. https://doi.org/10.1051/0004-6361/201322494

[Duffy 2012] Christopher Duffy, Yolanda Gil, Ewa Deelman, Suresh Marru, Marlon Pierce, Ibrahim Demir, and Gerry Wiener, **Designing a Road Map for Geoscience Workflows**, Eos, Vol. 93, No. 24, 12 June 2012, 225–226 https://doi.org/10.1029/2012EO240002

[Garijo 2014] Daniel Garijo, Pinar Alper, Khalid Belhajjame, Oscar Corcho, Yolanda Gil, Carole Goble, **Common motifs in scientific workflows: An empirical analysis**, (2014) Future Generation Computer Systems, 2014, 36 https://doi.org/10.1016/j.future.2013.09.018

[Chard 2016] K. Chard *et al.*, **I'll take that to go: Big data bags and minimal identifiers for exchange of large, complex datasets** *2016 IEEE International Conference on Big Data (Big Data)*, Washington, DC,2016,pp.319-328 https://doi.org/10.1109/BigData.2016.7840618

[Herschel 2017] Melanie Herschel, Ralf Diestelkämper, Houssem Ben Lahmar **A survey on provenance: What for? What form? What from?** The VLDB Journal Volume 26 Issue 6, December 2017 Pages 881-906 https://doi.org/10.1007/s00778-017-0486-1

[Wilkinson 2016] Mark D. Wilkinson, Michel Dumontier et al, **The FAIR Guiding Principles for scientific data management and stewardship**, Scientific Data volume 3, 2016, https://doi.org/10.1038/sdata.2016.18

[Katz 2014] Katz, D.S., 2014. Transitive Credit as a Means to Address Social and Technological Concerns Stemming from Citation and Attribution of Digital Products. Journal of Open Research Software, 2(1), p.e20. DOI: http://doi.org/10.5334/jors.be

[Garijo 2017] Daniel Garijo, Yolanda Gil, Oscar Corcho **Abstract, link, publish, exploit: An end to end framework, for workflow sharing**, Future Generation Computer Systems Volume 75, October 2017, Pages 271-283, https://doi.org/10.1016/j.future.2017.01.008

[Wroe 2006] Chris Wroe, Carole Goble, Antoon Goderis, Phillip Lord, Simon Miles, Juri Papay, Pinar Alper, Luc Moreau, **Recycling workflows and services through discovery and reuse**, CCPE, 11 May 2006, https://doi.org/10.1002/cpe.1050

[Alper 2018] Pinar Alper, Khalid Belhajjame, Vasa Curcin and Carole A. Goble, **LabelFlow Framework for Annotating Workflow Provenance**, Informatics 2018, 5(1), 11; https://doi.org/10.3390/informatics5010011

[Moreno 2019] Pablo Moreno, Luca Pireddu, Pierrick Roger, Nuwan Goonasekera, Enis Afgan, Marius van den Beek, Sijin He, Anders Larsson, Christoph Ruttkies, Daniel Schober, David Johnson, Philippe Rocca-Serra, Ralf J.M. Weber, Bjoern Gruening, Reza Salek, Namrata Kale, Yasset Perez-Riverol, Irene Papatheodorou, Ola Spjuth, Steffen Neumann, **Galaxy-Kubernetes integration: scaling bioinformatics workflows in the cloud**, bioRxiv 488643; doi: https://doi.org/10.1101/488643

[Starlinger 2014] Johannes Starlinger, Bryan Brancotte, Sarah Cohen-Boulakia, Ulf Leser, **Similarity search for scientific workflows** Proceedings of the VLDB Endowment Volume 7 Issue 12, August 2014 Pages 1143-1154, https://doi.org/10.14778/2732977.2732988

[Starlinger 2012] Starlinger J., Cohen-Boulakia S., Leser U. (2012) **(Re)Use in Public Scientific Workflow Repositories**. In: Ailamaki A., Bowers S. (eds) Scientific and Statistical Database Management. SSDBM 2012. Lecture Notes in Computer Science, vol 7338. https://doi.org/10.1007/978-3-642-31235-9_24

[Turner 2015] Kenneth J. Turner, Paul S. Lambert **Workflows for quantitative data analysis in the social sciences** International Journal on Software Tools for Technology Transfer June 2015, 17(3), pp 321–338 https://doi.org/10.1007/s10009-014-0315-4

[McPhillips 2015] Timothy McPhillips, Tianhong Song, Tyler Kolisnik, Steve Aulenbach, Khalid Belhajjame, Kyle Bocinsky, Yang Cao, Fernando Chirigati, Saumen Dey, Juliana Freire, Deborah Huntzinger, Christopher Jones, David Koop, Paolo Missier, Mark Schildhauer, Christopher Schwalm, Yaxing Wei, James Cheney, Mark Bieda, Bertram Ludaescher **YesWorkflow: A User-Oriented, Language-Independent Tool for Recovering Workflow Information from Scripts** (2015) International Journal of Digital Curation 10(1). https://doi.org/10.2218/ijdc.v10i1.370

[Haendel 2016] Melissa Haendel, Andrew Su, Julie McMurry, & et al. (2016, December 15). **FAIR-TLC: Metrics to Assess Value of Biomedical Digital Repositories:** Response to RFI NOT-OD-16-133. Zenodo. https://doi.org/10.5281/zenodo.203295