Reproducibility; Research Objects (RO-Crate) and Common Workflow Language (CWL)

Stian Soiland-Reyes

eScience lab, The University of Manchester

INDElab, University of Amsterdam

@soilandreyes

https://orcid.org/0000-0001-9842-9718

https://slides.com/soilandreyes/

H2020-INFRAEOSC-2018-2 824087
H2020-INFRAEDI-2018-1 823830
H2020-INFRAIA-2017-1 730976
H2020-INFRADEV-2019-2 871118
H2020-INFRAIA-2018-1 823827

@soilandreyes

WoSSS21: Workshop on Sustainable Software Sustainability

2021-10-07

WORLD VIEW
A personal take on events
https://doi.org/10.1038/d41586-019-01307-2

Rein in the four horsemen of irreproducibility

Dorothy Bishop *describes how threats to reproducibility, recognized but unaddressed for decades, might finally be brought under control.*

More than four decades into my scientific career, I find myself an outlier among academics of similar age and seniority: I strongly identify with the movement to make the practice of science more robust. It's not that my contemporaries are unconcerned about doing science well; it's just that many of them don't seem to recognize that there are serious problems with current practices. By contrast, I think that, in two decades, we will look back on the past 60 years — particularly in biomedical science — and marvel at how much time and money has been wasted on flawed research.

How can that be? We know how to formulate and test hypotheses in controlled experiments. We can account for unwanted variation with statistical techniques. We appreciate the need to replicate observations.

Yet many researchers persist in working in a way almost guaran-

MANY RESEARCHERS PERSIST IN WORKING IN A WAY ALMOST **GUARANTEED** NOT TO DELIVER **MEANINGFUL RESULTS.**

"They ride with what I refer to as the four horsemen of the reproducibility apocalypse:

1. Publication bias
2. Low statistical power
3. P-value hacking
4. HARKing (hypothesizing after results are known)

https://doi.org/10.1038/s41586-019-0965-1

Reproducibility?

Assignment of MAGs to reference databases

Four reference databases were used to classify the set of MAGs recovered from the human gut assemblies: HR, RefSeq, GenBank and a collection of MAGs from public datasets. HR comprised a total of 2,468 high-quality genomes (>90% completeness, <5% contamination) retrieved from both the HMP catalogue (https://www.hmpdacc.org/catalog/) and the HGG[8]. From the RefSeq database, we used all the complete bacterial genomes available (n = 8,778) as of January 2018. In the case of GenBank, a total of 153,359 bacterial and 4,053 eukaryotic genomes (3,456 fungal and 597 protozoan genomes) deposited as of August 2018 were considered. Lastly, we surveyed 18,227 MAGs from the largest datasets publicly available as of August 2018[13,16,17,18,19], including those deposited in the Integrated Microbial Genomes and Microbiomes (IMG/M) database[52]. For each database, the function 'mash sketch' from Mash v.2.0[53] was used to convert the reference genomes into a MinHash sketch with default *k*-mer and sketch sizes. Then, the Mash distance between each MAG and the set of references was calculated with 'mash dist' to find the best match (that is, the reference genome with the lowest Mash distance). Subsequently, each MAG and its closest relative were aligned with dnadiff v.1.3 from MUMmer 3.23[54] to compare each pair of genomes with regard to the fraction of the MAG aligned (aligned query, AQ) and ANI.

Technical reproducibility and cluster quality

A random subset of 1,000 metagenomes (Supplementary Table 1) was tested with two additional approaches to assess the reproducibility of the MAGs generated here. With one of the methods, metagenomes were assembled with MEGAHIT v.1.1.3[24] and subsequently binned with MetaBAT 2, MetaBAT 1 and MaxBin v.2.2.4[62]. A refinement step was then performed using the bin_refinement module from MetaWRAP v.1.0[25] to combine and improve the results generated by the three binners. The second method involved a modified co-assembly approach, in which individual assemblies from the same study were first merged and dereplicated with CD-HIT v.4.7[63] (cd-hit-est with option -c 0.99 defining a sequence identity threshold of 99%). Metagenomic datasets were then mapped to their merged, non-redundant assembly with BWA-MEM to obtain co-abundance information for binning with MetaBAT 2 (with option --minContig 2000). The resulting MAGs with a QS >50 obtained with each method were compared to the MAGs recovered with our main pipeline (individual assembly with SPAdes, plus binning with MetaBAT 2) for the same 1,000 datasets, using the combined Mash and MUMmer workflow described above.

Read classification of the 13,133 human gut metagenomic datasets was performed with sourmash v.2.0.0a4[68] against the HR, RefSeq and UMGS genome collections. Signature files were generated for both the reference (FASTA) and query (FASTQ) files, with 'sourmash compute --scaled 1000 -k 31 --track-abundance'. For each set of references, a lowest common ancestor database was created ('sourmash lca index --scaled 1000 -k 31'), with each genome representing a unique species lineage. Raw reads were then compared with 'sourmash lca gather' against each database. Species prevalence and abundance was determined with BWA-MEM, where species presence was inferred by assessing the level of genome coverage, mean read depth and depth evenness. First, we calculated depth and variation penalty scores corresponding to the missing coverage (100% – genome coverage) multiplied by either the log(mean depth) or the depth coefficient of variation (defined as the standard deviation of read depth divided by the mean), respectively. These metrics allowed us to gauge both coverage and depth simultaneously, as genomes that have a high mean depth (or high depth variation) but are not well covered are less likely to be present in the sample than those that have the same level of coverage with lower read depth. Thresholds for determining genome presence were set at a minimum coverage of at least 60%, and both depth and variation penalty scores at a maximum of the 99th percentile

De novo assembly and binning

Raw reads from each run were first assembled with SPAdes v.3.10.0[20] with option --meta[21]. Thereafter, MetaBAT 2[15] (v.2.12.1) was used to bin the assemblies using a minimum contig length threshold of 2,000 bp (option --minContig 2000) and default parameters. Depth of coverage required for the binning was inferred by mapping the raw reads back to their assemblies with BWA-MEM v.0.7.16[45] and then calculating the corresponding read depths of each individual contig with samtools v.1.5[46] ('samtools view -Sbu' followed by 'samtools sort') together with the jgi_summarize_bam_contig_depths function from MetaBAT 2. The QS of each metagenome-assembled genome (MAG) was estimated with CheckM v.1.0.7[22] using the lineage_wf workflow and calculated as: level of completeness – 5 × contamination. Ribosomal RNAs (rRNAs) were detected with the cmsearch function from INFERNAL v.1.1.2[47] (options -Z 1000 --hmmonly --cut_ga) using the Rfam[48] covariance models of the bacterial 5S, 16S and 23S rRNAs. Total alignment length was inferred by the sum of all non-overlapping hits. Each gene was considered present if more than 80% of the expected sequence length was contained in the MAG. Transfer RNAs (tRNAs) were identified with tRNAscan-s.e. v.2.0[49] using the bacterial tRNA model (option -B) and default parameters. Classification into high- and medium-quality

https://github.com/Finn-Lab/MGS-gut

Finn-Lab / MGS-gut

Watch 4

Code Issues Pull requests Actions Projects Security Insights

master 2 branches 0 tags Go to file Code

alexmsalmeida Update LICENSE 649913e on 12 Nov 122 commits

R	Update funcs_phy-assoc_fig5b.R	13 months ago
pipelines	Update map2ref.sh	2 years ago
scripts	Update parse_checkm.py	2 years ago
LICENSE	Update LICENSE	last month
README.md	Update README.md	13 months ago

README.md

Analysis of Metagenomic Species (MGS)

Scripts used for characterizing metagenome-assembled genomes (MAGs) used in the following publication:

A Almeida, AL Mitchell, M Boland, SC Forster, GB Gloor, A Tarkowska, TD Lawley and RD Finn (2019) A new genomic blueprint of the human gut microbiota. *Nature* **568**, 499–504

Associated data can also be found in our FTP server.

mashdiff.sh

Compare a set of genomes against a reference database. Selects best representative for whole-genome alignment.

Requirements:

- Mash (tested v2.0)
- MUMmer (tested v3.23)
- Python 2.7

Coded for running within LSF cluster environments.

Usage:

```
mashdiff.sh -i genome_folder/ -r reference.msh -s db_name -p output_prefix
```

About

Analysing Metagenomic Species (MGS)

Readme

View license

Releases

No releases published

Packages

No packages published

Languages

R 70.7% Python 18.5%

Shell 10.8%

Why use workflows?

Automation

- Automate computational aspects
- Repetitive pipelines, sweep campaigns

Scaling—compute cycles

- Make use of computational infrastructure
- Handle large data

Abstraction—people cycles

- Shield complexity and incompatibilities
- Report, re-usue, evolve, share, compare
- Repeat—Tweak—Repeat
- First-class commodities

Provenance—reporting

- Capture, report and utilize log and data lineage
- Auto-documentation
- Tracable evolution, audit, transparency
- Reproducible science

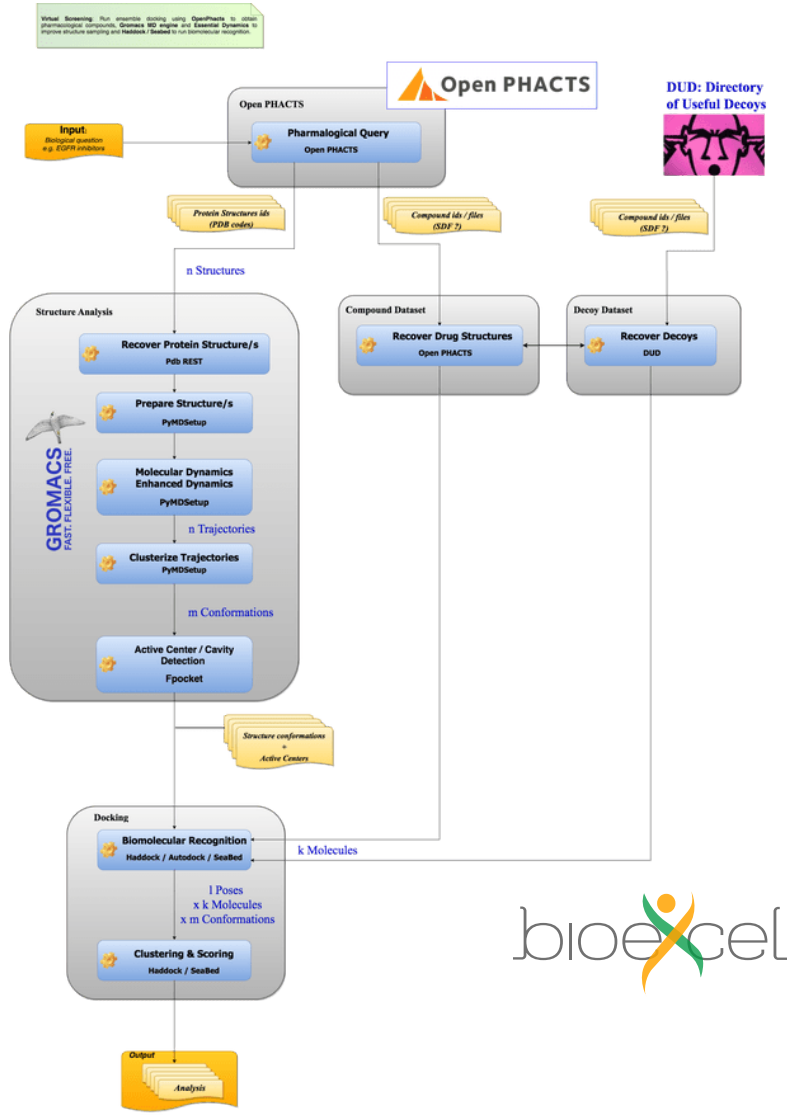
Findable

Accessible

Interoperable

Reusable

(Reproducible)



bioexcel

Adapted from Bertram Ludäscher (2015)

https://www.slideshare.net/ludaesch/works-2015provenancemileage

https://doi.org/10.1007/s13222-012-0100-z

Existing Workflow systems

Ryan edited this page 3 days ago · 299 revisions

Permalink: https://s.apache.org/existing-workflow-systems

Cite as (update dates):

Peter Amstutz, Maxim Mikheev, Michael R. Crusoe, Nebojša Tijanić, Samuel Lampa, et al. (2021): **Existing Workflow systems.** *Common Workflow Language wiki*, GitHub. https://s.apache.org/existing-workflow-systems updated 2021-06-29, accessed 2021-09-03.

Computational Data Analysis Workflow Systems

An incomplete list

Please add new entries at the bottom.

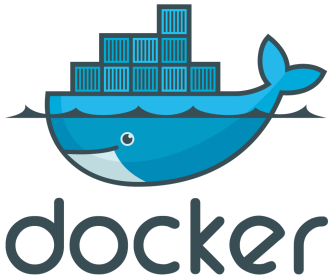
See also: https://github.com/pditommaso/awesome-pipeline

1. Arvados - CWL-based distributed computing platform for data analysis on massive data se https://github.com/arvados/arvados
2. Apache Taverna http://www.taverna.org.uk/ https://taverna.incubator.apache.org/
3. Galaxy http://galaxyproject.org/
4. SHIWA https://www.shiwa-workflow.eu/
5. Apache Oozie https://oozie.apache.org/
6. DNANexus https://wiki.dnanexus.com/API-Specification-v1.0.0/IO-and-Run-Specifications Specification-v1.0.0/Workflows-and-Analyses
7. ~~BioDT~~ http://www.biodatomics.com/ (dead link)
8. Agave http://agaveapi.co/live-docs/
9. DiscoveryEnvironment http://www.iplantcollaborative.org/ci/discovery-environment
10. Wings http://www.wings-workflows.org/
11. Knime https://www.knime.org/

291. BRANE Framework https://onnovalkering.github.io/brane/
292. ApolloWF https://apollowf.github.io/
293. IS-EPOS Platform https://ieeexplore.ieee.org/document/9308147 https://tcs.
294. pyinvoke http://www.pyinvoke.org/
295. targets R package https://cran.r-project.org/package=targets
296. Compi https://doi.org/10.7717/peerj-cs.593 https://github.com/sing-group/
297. TriggerFlow: Event-based Orchestration of Serverless Workflows (https://gith
298. Google Cloud Workflows: Orchestrate and automate Google Cloud and HTT workflows. (https://cloud.google.com/workflows/docs)
299. PanDA Workflow Management System: https://doi.org/10.1051/epjconf/201 /201610801003 https://github.com/PanDAWMS
300. Harvester: https://github.com/HSF/harvester

https://s.apache.org/existing-workflow-systems

COMMON WORKFLOW LANGUAGE



Implementations

Software	Description
cwltool	Reference implementation of CWL
Arvados	Distributed computing platform for data analysis on massive data sets. Using CWL on Arvados
Toil	Toil is a workflow engine entirely written in Python.
Apache Taverna	Domain-independent Workflow Management System
Galaxy	Web-based platform for data intensive biomedical research.
AWE	Workflow and resource management system for bioinformatics data analysis.
Funnel	Use Google Genomics Pipeline API with CWL
Rabix Bunny	Reproducible Analyses for Bioinformatics

https://www.commonwl.org/#Implementations

```
cwlVersion: v1.0
class: Workflow
inputs:
  inp: File
  ex: string

outputs:
  classout:
    type: File
    outputSource: compile/classfile

steps:
  untar:
    run: tar-param.cwl
    in:
      tarfile: inp
      extractfile: ex
    out: [example_out]

  compile:
    run: arguments.cwl
    in:
      src: untar/example_out
    out: [classfile]
```

http://www.commonwl.org/

MENU nature International journal of science

TOOLBOX · 02 SEPTEMBER 2019

Workflow systems turn raw data into scientific knowledge

How workflow tools can make your computational methods portable, maintainable, reproducible and shareable.

BY JEFFREY M. PERKEL

Reinventing the wheel is pointless, but for computational biologists it's sometimes unavoidable. So when Rob Finn and Folker Meyer realized how much their work overlapped, they decided to try something different.

Finn is head of the sequence-families team at the European Bioinformatics Institute (EBI) in Hinxton, UK; Meyer is a computer scientist at Argonne National Laboratory in Lemont, Illinois. Both run facilities that let researchers perform a computationally intensive process called metagenomic analysis, which allows microbial communities to be reconstructed from shards of DNA. It would be helpful, they realized, if they could try each other's code. The problem was that their analytical 'pipelines' — the carefully choreographed computational steps required to turn raw data into scientific knowledge — were written in different languages. Meyer's team was using an in-house system called AWE, whereas Finn was working with nearly 9,500 lines of Python code.

"It was a horrible Python code base," says Finn — complicated, and difficult to maintain. "Bits had been bolted on in an ad hoc fashion over seven years by at least four different developers." And it was "heavily tied to the compute infrastructure", he says, meaning it was written for specific computational resources and a particular way of organizing fi
essentially unusable outside the
the EBI wasn't using AWE, the rev
true. Then Finn and Meyer lear
Common Workflow Language (

CWL is a way of describing an
lines and computational tools —
than 250 systems now available, in
popular options as Snakemake, N
Galaxy. Although they speak differ
and support different features, t
have a common aim: to make co
methods reproducible, portable,
and shareable. CWL is essentially
language that researchers can use to share pipe-
lines for whichever system. For Finn, that

language brought sanity to his codebase, reducing it by around 73%. Importantly, it has made it easier to test, execute and share new methods, and to run them on the cloud.

There is a learning curve to adopting workflow languages. But, says Brian Naughton, data lead and co-founder of the drug-discovery firm Hexagon Bio in Menlo Park, California, "the energy that you expend learning is more than made up for by the energy you save in having your code be reproducible."

Nature **573**, 149-150 (2019)
https://doi.org/10.1038/d41586-019-02619-z

https://github.com/NCBI-Hackathons/Epigenomics_CWL/blob/master/cwl/tools/preprocess.cwl

```yaml
cwlVersion: v1.0
class: Workflow

inputs:
  toConvert: File


outputs:
  converted:
    type: File
    outputSource: convertMethylation/converted
  combined:
    type: File
    outputSource: mergeSymmetric/combined


steps:
  convertMethylation:
    run: interconverter.cwl
    in:
      toConvert: toConvert
    out: [converted]
  mergeSymmetric:
    run: symmetriccpgs.cwl
    in:
      toCombine: convertMethylation/converted
    out: [combined]
```

```yaml
cwlVersion: v1.0
class: CommandLineTool
inputs:
  toConvert:
    type: File
    inputBinding:
      prefix: -i
outputs:
  converted:
    type: File
    outputBinding:
      glob: "*.meth"
baseCommand: interconv
arguments: ["-d", $(ru

hints:
  - class: DockerRequi
    dockerPull: "quay.
```

```yaml
cwlVersion: v1.0
class: CommandLineTool
inputs:
  toCombine:
    type: File
    inputBinding:
      prefix: -i
outputs:
  combined:
    type: File
    outputBinding:
      glob: "*.sym"

baseCommand: symmetriccpgs.sh
arguments: ["-d", $(runtime.outdir)]

hints:
  - class: DockerRequirement
    dockerPull: "quay.io/neksa/screw-tool"
```

@soilandreyes

https://github.com/NCBI-Hackathons/Epigenomics_CWL/blob/master/cwl/tools/preprocess.cwl

```yaml
cwlVersion: v1.0
class: Workflow

inputs:
  toConvert: File


outputs:
  converted:
    type: File
    outputSource: convertMethylation/converted
  combined:
    type: File
    outputSource: mergeSymmetric/combined


steps:
  convertMethylation:
    run: interconverter.cwl
    in:
      toConvert: toConvert
    out: [converted]
  mergeSymmetric:
    run: symmetriccpgs.cwl
    in:
      toCombine: convertMethylation/converted
    out: [combined]
```

```yaml
cwlVersion: v1.0
class: CommandLineTool
inputs:
  toConvert:
    type: File
    inputBinding:
      prefix: -i
outputs:
  converted:
    type: File
    outputBinding:
      glob: "*.meth"
baseCommand: interconv
arguments: ["-d", $(ru

hints:
  - class: DockerRequi
    dockerPull: "quay.
```

```yaml
cwlVersion: v1.0
class: CommandLineTool
inputs:
  toCombine:
    type: File
    inputBinding:
      prefix: -i
outputs:
  combined:
    type: File
    outputBinding:
      glob: "*.sym"

baseCommand: symmetriccpgs.sh
arguments: ["-d", $(runtime.outdir)]

hints:
  - class: DockerRequirement
    dockerPull: "quay.io/neksa/screw-tool"
```

@soilandreyes

https://github.com/NCBI-Hackathons/Epigenomics_CWL/blob/master/cwl/tools/preprocess.cwl

```yaml
cwlVersion: v1.0
class: Workflow

inputs:
  toConvert: File


outputs:
  converted:
    type: File
    outputSource: convertMethylation/converted
  combined:
    type: File
    outputSource: mergeSymmetric/combined


steps:
  convertMethylation:
    run: interconverter.cwl
    in:
      toConvert: toConvert
    out: [converted]
  mergeSymmetric:
    run: symmetriccpgs.cwl
    in:
      toCombine: convertMethylation/converted
    out: [combined]
```

```yaml
cwlVersion: v1.0
class: CommandLineTool
inputs:
  toConvert:
    type: File
    inputBinding:
      prefix: -i
outputs:
  converted:
    type: File
    outputBinding:
      glob: "*.meth"
baseCommand: interconv
arguments: ["-d", $(ru

hints:
  - class: DockerRequi
    dockerPull: "quay.
```

```yaml
cwlVersion: v1.0
class: CommandLineTool
inputs:
  toCombine:
    type: File
    inputBinding:
      prefix: -i
outputs:
  combined:
    type: File
    outputBinding:
      glob: "*.sym"

baseCommand: symmetriccpgs.sh
arguments: ["-d", $(runtime.outdir)]

hints:
  - class: DockerRequirement
    dockerPull: "quay.io/neksa/screw-tool"
```

@soilandreyes

https://github.com/NCBI-Hackathons/Epigenomics_CWL/blob/master/cwl/tools/preprocess.cwl

```yaml
cwlVersion: v1.0
class: Workflow

inputs:
  toConvert: File


outputs:
  converted:
    type: File
    outputSource: convertMethylation/converted
  combined:
    type: File
    outputSource: mergeSymmetric/combined


steps:
  convertMethylation:
    run: interconverter.cwl
    in:
      toConvert: toConvert
    out: [converted]
  mergeSymmetric:
    run: symmetriccpgs.cwl
    in:
      toCombine: convertMethylation/converted
    out: [combined]
```

```yaml
cwlVersion: v1.0
class: CommandLineTool
inputs:
  toConvert:
    type: File
    inputBinding:
      prefix: -i
outputs:
  converted:
    type: File
    outputBinding:
      glob: "*.meth"
baseCommand: interconv
arguments: ["-d", $(ru

hints:
  - class: DockerRequi
    dockerPull: "quay.
```

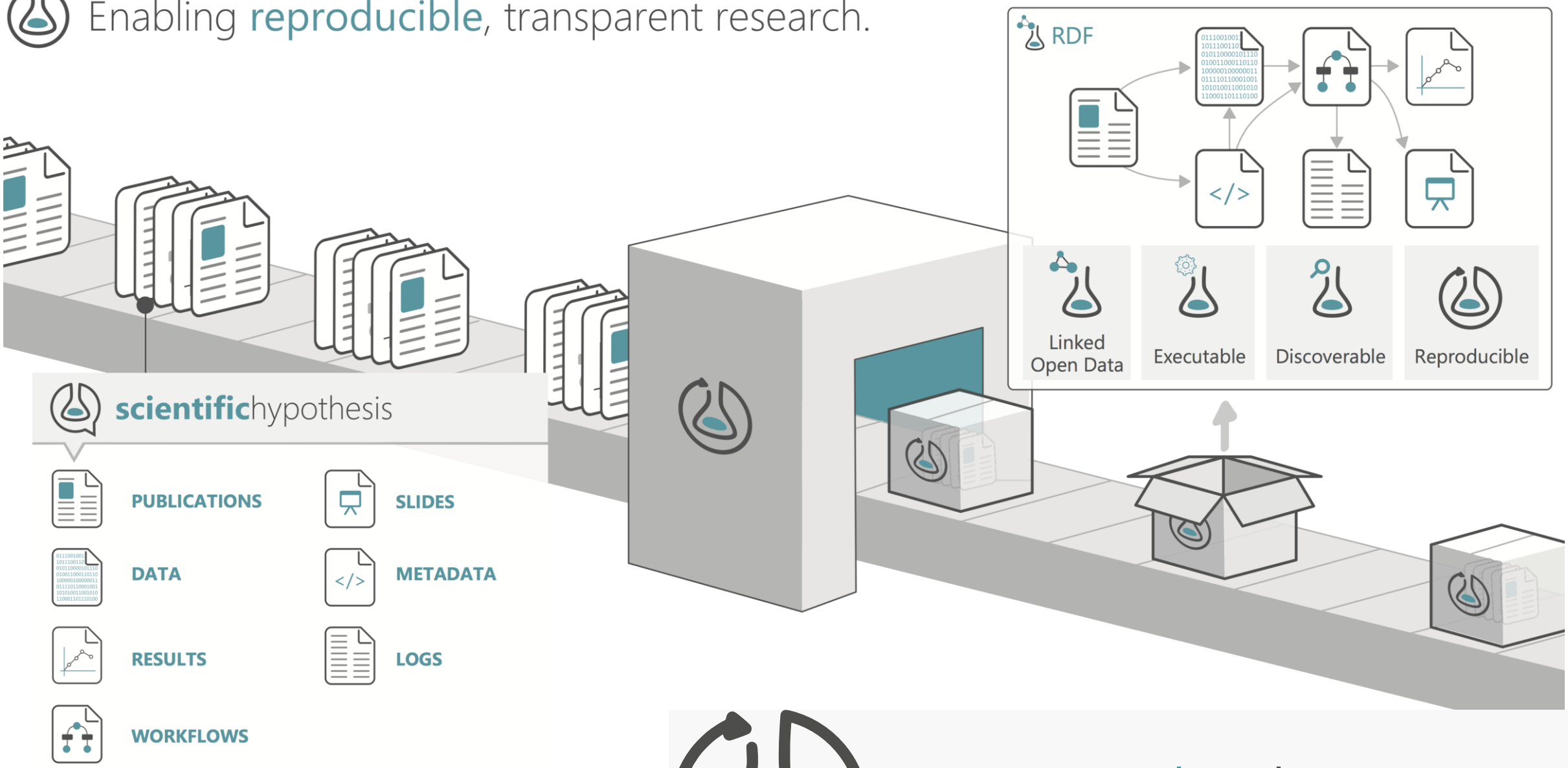
```yaml
cwlVersion: v1.0
class: CommandLineTool
inputs:
  toCombine:
    type: File
    inputBinding:
      prefix: -i
outputs:
  combined:
    type: File
    outputBinding:
      glob: "*.sym"

baseCommand: symmetriccpgs.sh
arguments: ["-d", $(runtime.outdir)]

hints:
  - class: DockerRequirement
    dockerPull: "quay.io/neksa/screw-tool"
```

@soilandreyes

Enabling reproducible, transparent research.



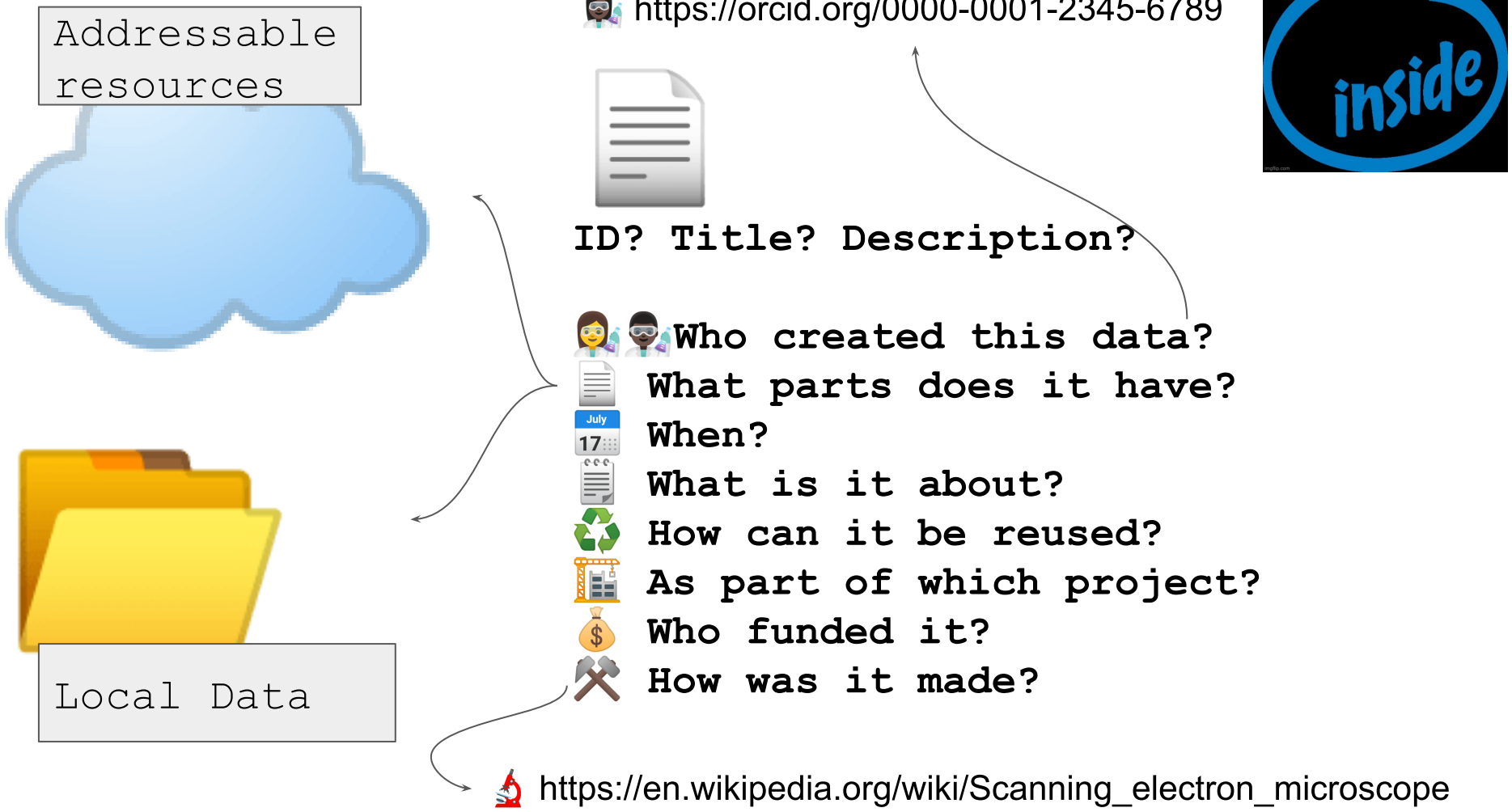
researchobject.org

https://doi.org/10.1016/j.future.2011.08.004

@soilandreyes

What is RO-Crate?

https://www.researchobject.org/ro-crate/



RO-Crate is method for self-decribed *datasets*
as a *digital object* using a single Linked Data
metadata document

Credit: Peter Sefton

Adapted from https://arkisto-platform.github.io/standards/ro-crate/

The dataset may contain *any kind* of resource, about anything, in *any format* as a file, URL or PID

Credit: Peter Sefton

Adapted from https://arkisto-platform.github.io/standards/ro-crate/

```
|-- Folder1/
|           |-- file1.this
|           |-- file2.that
|-- Folder2/
|            -- file1.this
|           |-- file2.that
|-2021-04-08 07.58.17.jpg
```

schema.org

```
{
    "@id": "2021-04-08 07.58.17.jpg",
    "@type": "File",
    "contentSize": 3271409,
    "dateModified": "2021-04-08T07:58:17+10:00",
    "description": "",
    "encodingFormat": [
      {"@id":
"https://www.nationalarchives.gov.uk/PRONOM/x-fmt/391"},
      "image/jpeg"
    ],
   "name": "Cute puppy"
},
```

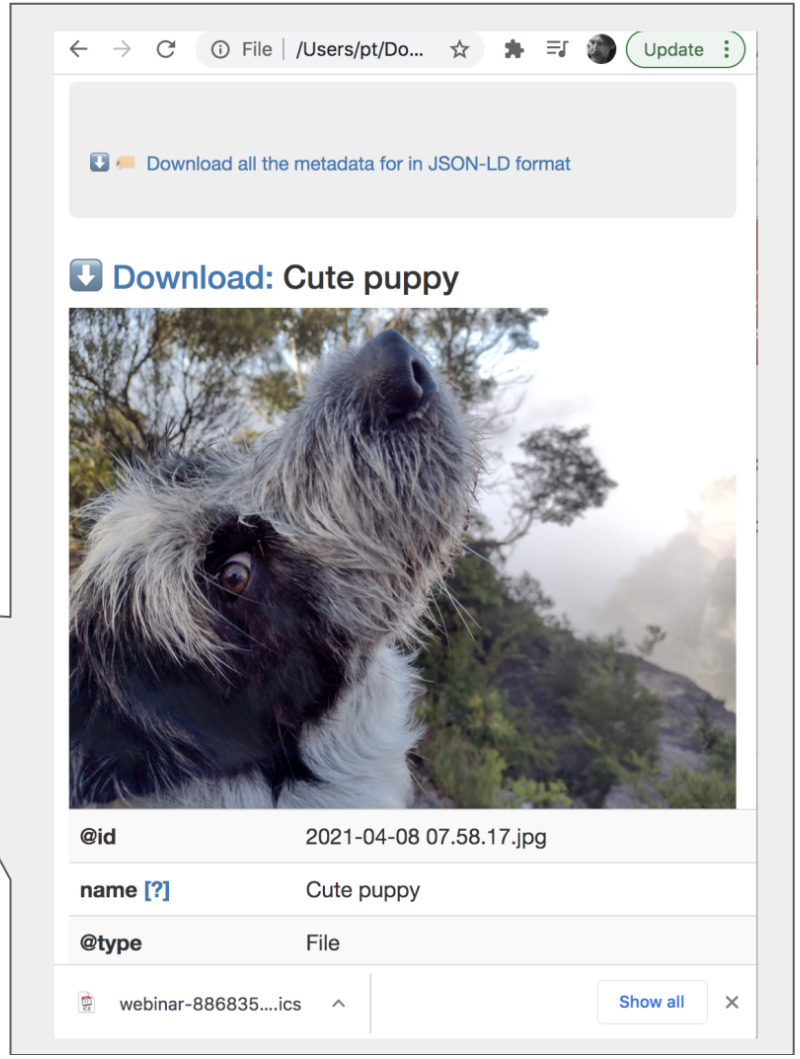
JSON-LD

Each resource have a **machine readable** description in JSON-LD format

Credit: Peter Sefton

Adapted from https://arkisto-platform.github.io/standards/ro-crate/

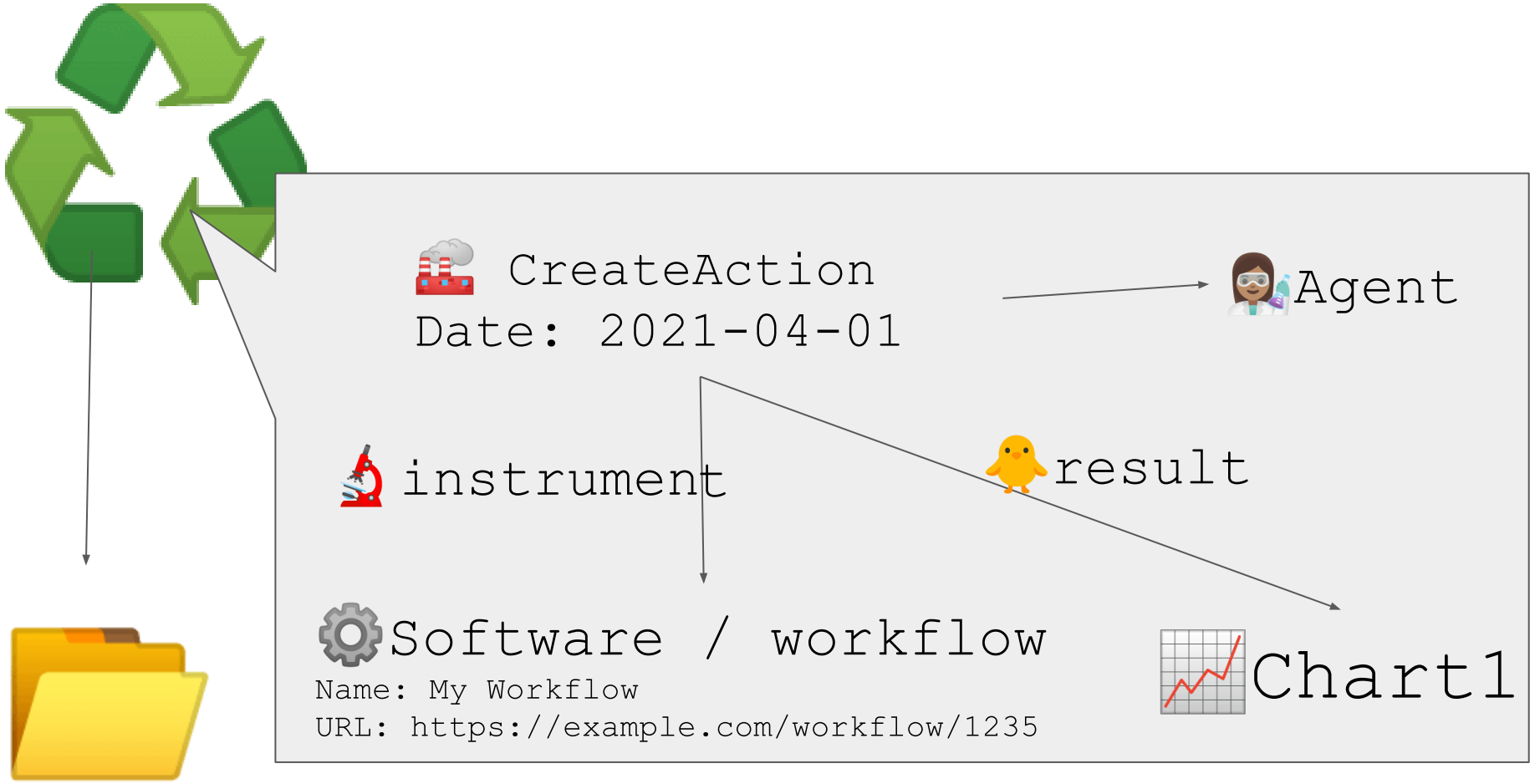
```
|-- Folder1/
|          |-- file1.this
|          |-- file2.that
|-- Folder2/
|          |-- file1.this
|          |-- file2.that
|-2021-04-08 07.58.17.jpg
```



A **human-readable** description/preview in an **HTML** file that *lives alongside the metadata*

Credit: Peter Sefton

Adapted from https://arkisto-platform.github.io/standards/ro-crate/



Provenance and **workflow** information can be included
– to assist in *re-use* of data and research processes

Credit: Peter Sefton

Adapted from https://arkisto-platform.github.io/standards/ro-crate/

RO-Crate Digital Objects may be *packaged* for distribution eg via Zip, Bagit and OCFL – or simply be *published on the Web*

Credit: Peter Sefton

Adapted from https://arkisto-platform.github.io/standards/ro-crate/

Techie deep-dive!

Warning: JSON ahead

RO-Crate Structure

The structure an *RO-Crate* MUST follow is:

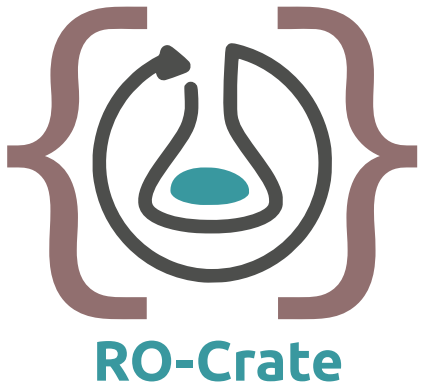
```
<RO-Crate root directory>/
|   ro-crate-metadata.json     # RO-Crate Metadat
|   ro-crate-preview.html      # RO-Crate Website
|   ro-crate-preview_files/    # MAY be present
|    | [other RO-Crate Website files]
|   [payload files and directories]  # 0 or more
```

Minimal example of RO-Crate

The following *RO-Crate Metadata File* represents a minimal description of an *RO-*

```
{ "@context": "https://w3id.org/ro/crate/1.1/context",
  "@graph": [

 {
    "@type": "CreativeWork",
    "@id": "ro-crate-metadata.json",
    "conformsTo": {"@id": "https://w3id.org/ro/crate/1.1"},
    "about": {"@id": "./"}
 },
 {
    "@id": "./",
    "identifier": "https://doi.org/10.4225/59/59672c09f4a4b",
    "@type": "Dataset",
    "datePublished": "2017",
    "name": "Data files associated with the manuscript:Effects of facil
    "description": "Palliative care planning for nursing home residents
    "license": {"@id": "https://creativecommons.org/licenses/by-nc-sa/3
 },
 {
  "@id": "https://creativecommons.org/licenses/by-nc-sa/3.0/au/",
  "@type": "CreativeWork",
  "description": "This work is licensed under the Creative Commons Attr
  "identifier": "https://creativecommons.org/licenses/by-nc-sa/3.0/au/"
  "name": "Attribution-NonCommercial-ShareAlike 3.0 Australia (CC BY-NC
 }
 ]
}
```



https://w3id.org/ro/crate/1.1

JSON for Linking Data

Data is messy and disconnected. JSON-LD organizes and connects it, creating a better Web.

https://json-ld.org/

⇄ Linked Data

Linked Data empowers people that publish and use information on the Web. It is a way to create a network of standards-based, machine-readable data across Web sites. It allows an application to start at one piece of Linked Data, and follow embedded links to other pieces of Linked Data that are hosted on different sites across the Web.

A Simple Example

```
{
  "@context": "http://json-ld.org/contexts/person.jsonld",
  "@id": "http://dbpedia.org/resource/John_Lennon",
  "name": "John Lennon",
  "born": "1940-10-09",
  "spouse": "http://dbpedia.org/resource/Cynthia_Lennon"
}
```

{⋔} JSON-LD

JSON-LD is a lightweight Linked Data format. It is easy for humans to read and write. It is based on the already successful JSON format and provides a way to help JSON data interoperate at Web-scale. JSON-LD is an ideal data format for programming environments, REST Web services, and unstructured databases such as CouchDB and MongoDB.

https://www.w3.org/TR/json-ld/

schema.org

Home

https://schema.org/

Dataset

Thing > CreativeWork > Dataset

A body of structured information describing some topic(s) of interest.

Usage: Between 100 and 1000 domains

Property	Expected Type	Description
Properties from Dataset		
distribution	DataDownload	A downloadable form of this dataset, at a specific location, in a specific format.
includedInDataCatalog	DataCatalog	A data catalog which contains this dataset. Supersedes catalog, includedDataCatalog. Inverse property: dataset.
Properties from CreativeWork		
about	Thing	The subject matter of the content.
accessibilityAPI	Text	Indicates that the resource is compatible with the referenced accessibility API (WebSchemas
accessibilityControl	Text	Identifies input methods that are sufficient to fully control the described resource (WebSchem values).
accessibilityFeature	Text	Content features of the resource, such as accessible media, alternatives and supported enh (WebSchemas wiki lists possible values).

Bioschemas

https://bioschemas.org/

Metadata

```
{
 "@id": "figure.png",
 "@type": ["File", "ImageObject"],
 "name": "XXL-CT-scan of an XXL Tyrannosaurus rex skull",
 "identifier": "https://doi.org/10.5281/zenodo.3479743",
 "author": {"@id": "https://orcid.org/0000-0002-8367-6908"},
 "encodingFormat": "image/png"
}
```

```
{
 "@id": "https://orcid.org/0000-0002-8367-6908",
 "@type": "Person",
 "affiliation": { "@id": "https://ror.org/03f0f6041" },
 "name": "J. Xuan"
}
```

ORCID

```
{
 "@id": "https://ror.org/03f0f6041",
 "@type": "Organization",
 "name": "University of Technology Sydney",
 "url": "https://www.uts.edu.au/"
}
```

ROR

Data and **Contextual** entities
described *within* RO-Crate Metadata File

Base vocabulary & types: **schema.org**

Cross-references to further contextual entities

RO-Crate **principle**:
Reuse existing PIDs and URLs

..but always **describe entities** which lack a human-readable resolution

Tooling!

How can I use it?

While we're mostly focusing on the specification, some tools already exist for working with RO-Crates:

- Describo interactive **desktop application** to create, update and export RO-Crates for different profiles. (~ *beta*)
- CalcyteJS is a **command-line** tool to help create RO-Crates and HTML-readable rendering (~ *beta*)
- ro-crate - **JavaScript**/NodeJS library for RO-Crate rendering as HTML. (~ *beta*)
- ro-crate-js - utility to render HTML from RO-Crate (~ *alpha*)
- ro-crate-ruby **Ruby** library to consume/produce RO-Crates (~ *alpha*)
- ro-crate-py **Python library** to consume/produce RO-Crates (~ *planning*)

These applications use or expose RO-Crates:

- Workflow Hub imports and exports Workflow RO-Crates
- OCFL-indexer NodeJS application that walks the Oxford Common File Layout on the file system, validate RO-Crate Metadata Files and parse into objects registered in Elasticsearch. (~ *alpha*)
- ONI indexer
- ocfl-tools
- ocfl-viewer
- Research Object Composer is a REST API for gradually building and depositing Research Objects according to a pre-defined profile. (RO-Crate support *alpha*)

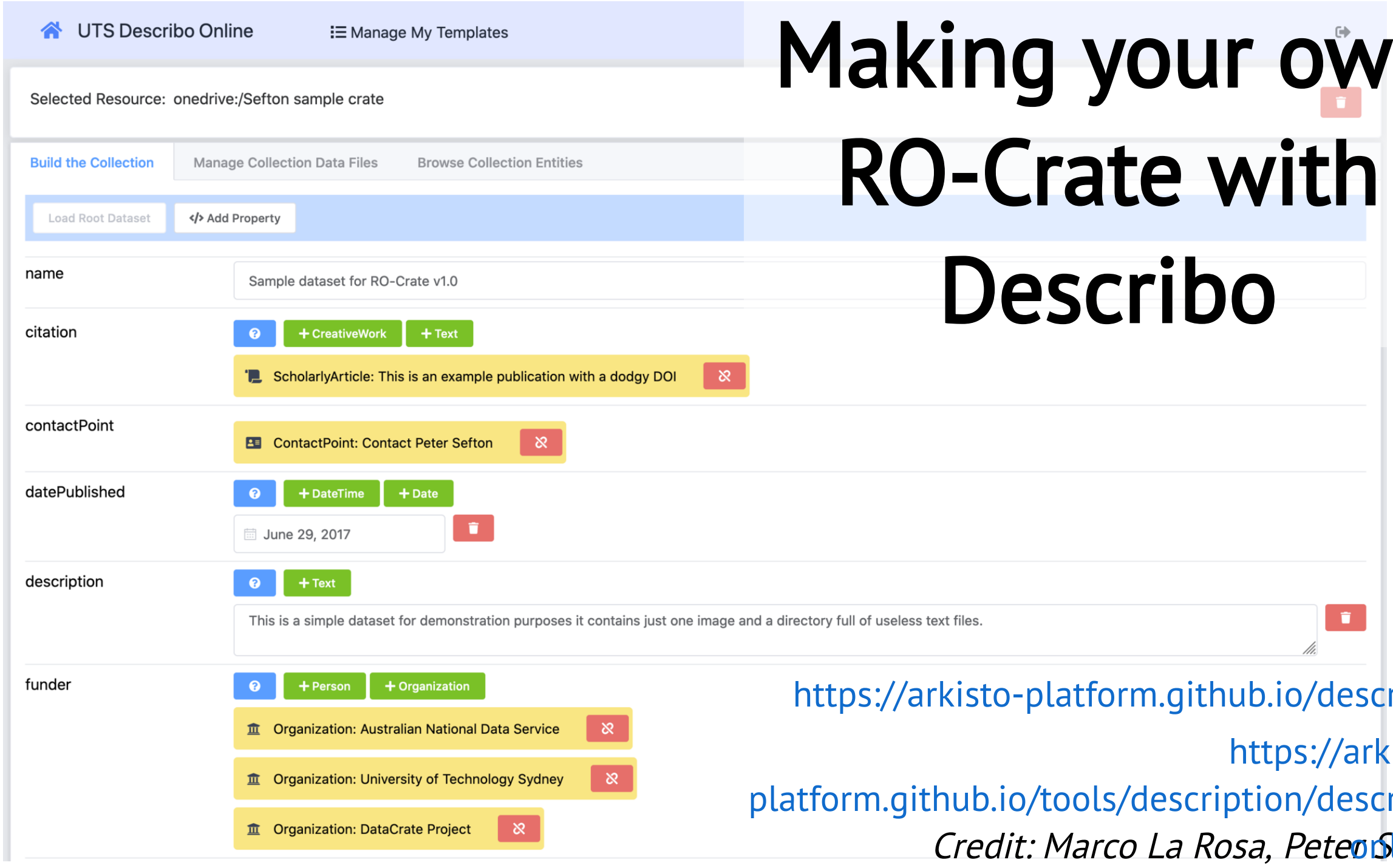
https://www.researchobject.org/ro-crate/implementations.html @soilandreyes

UNIVERSITY OF TECHNOLOGY, SYDNEY

RO-Crate for data scientists

@soilandreyes

Making your own RO-Crate with Describo



https://arkisto-platform.github.io/describo/

https://arkisto-platform.github.io/tools/description/describo-online/

Credit: Marco La Rosa, Peter Sefton

Dataset: Survey of Victoria Arch, Wombeyan Caves NSW

https://doi.org/10.4225/59/5a4d9b76d79f4

https://data.research.uts.edu.au/examples/ro-crate/0.2/Victoria_Arch_pub/

Download this Dataset

Download all the metadata for Survey of Victoria Arch, Wombeyan Caves NSW in JSON-LD format

Check this crate

FAIR is not just machine-readable!

@id	https://dx.doi.org/10.4225/59/5a4d9b76d79f4
name	Survey of Victoria Arch, Wombeyan Caves NSW
@type	Dataset
description	This data is part of a project by Michael Lake and supported by the Australian Speleological Federation. Data was acquired at Wombeyan Caves by Robert Zlot in January 2014 using the Zebedee 3D Mapping System developed by CSIRO.
datePublished	2017-10-01
creator	• Robert Zlot • Mike Lake • Lukas Kaul
path	./
contactPoint	Contact Mike Lake
contentLocation	Victoria Arch
hasPart	README.txt--to--wcc06_archencentre1.laz wcc04_archentrance.laz--to--README.html
identifier	• https://dx.doi.org/10.4225/59/5a4d9b76d79f4 • dx.doi.org/10.4225/59/5a4d9b76d79f4
license	Copyright CSIRO 2014: Available for research and academic purposes

https://data.research.uts.edu.au/examples/ro-crate/0.2/Victoria_Arch_pub/ro-crate-preview.html

RO-Crate for digital humanities

Capturing cultural heritage records as RO-Crates

Home | Advanced Search | Transcription Search

Show OCFL inventory file | Show RO-Crate | Show Objectified Crate | Show Data files

Versions: v1

Recordings in South Efate

Open (subject to agreeing to PDSC access conditions)

Item: NT1 / 98007
Collection: NT1
Contributors:

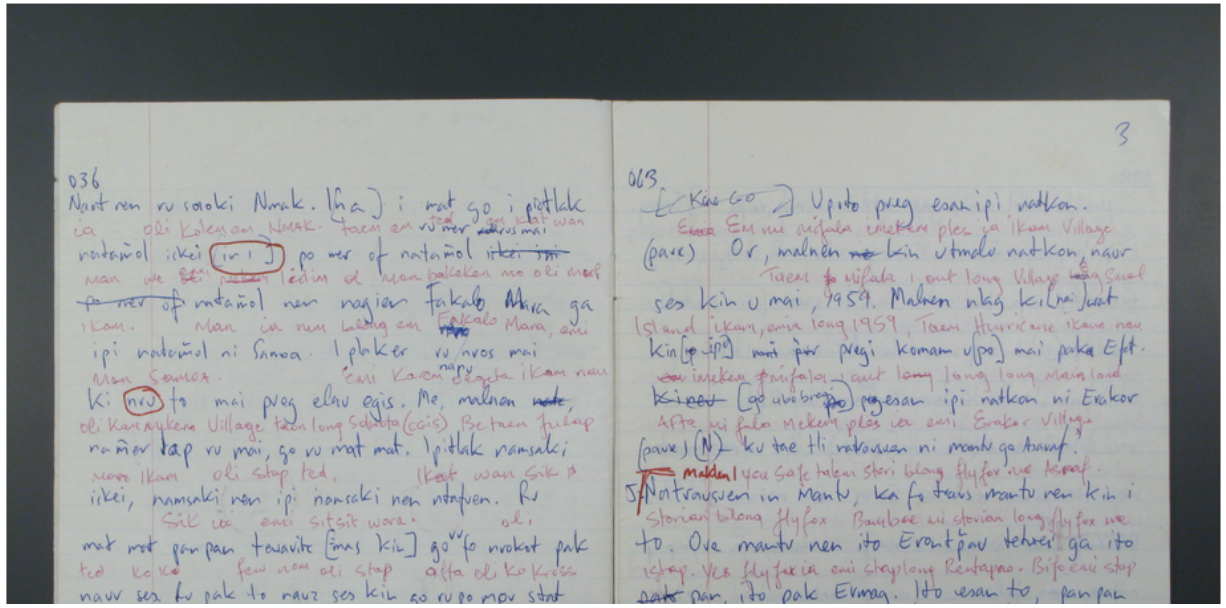
- Nick Thieberger (collector)
- Corin Bone (operator)
- Nick Thieberger (recorder)
- Kalsarap Namaf (speaker)
- Iokopeth (speaker)
- John Maklen (speaker)
- Waia Tenene (speaker)

NT1-98007. Text #047 (speaker is John Maklen. Text title: History of villages before Erakor); Text #048 (speaker is John Maklen. Text title: Mantu the flying fox and Erromango); Text #049. Text title: Asaraf (speaker is John Maklen);Text #050. Text title: Mumu and Kotkot (speaker is John Maklen); Text #051. Text title: Natopu ni Erakor—the spirit who lives at Erakor (speaker is John Maklen);Text #038. Text title: The need for respect (speaker is Iokopeth) Stories can be seen at NT8-TEXT. There are time-aligned transcripts of this item and handwritten transcripts by Manuel Wayane scanned as jpg files. show less

show item metadata

Content Languages

PARADISEC



Images | Audio | XML Files

NT1-98007-98007A

0:00 / 45:29

NT1-98007-98007A.ixt | NT1-98007-98007A.eaf

▶ PLAY (09:22)

Natopu nen kin ito esan ga, ipi natopu nigmam.
The natopu (local spirit) that lives here, it is our natopu.

Natopu	nen	kin	ito	esan	ga,	ipi	natopu	nigmam.
natopu	nen	kin	i= to	esan	ga	i= pi	natopu	nigmam
spirit	that	COMP	3S.RS= stay	place	3S.	3S.RS= be	spirit	1P.POS

▶ PLAY (09:26)

Ser natamõl ni Erakor runomser mtaki natopu nen kin rusoso ki maarik.
Everyone from Erakor is scared of the natopu which they call 'maarik' (mister)

https://arkisto-platform.github.io/case-

@sc

https://mod.paradisec.org.au/view/NT1/98007?version=v1

http://mod.paradisec.org.au/view/NT1/98007

RO-Crate for repositories

RO-Crate as a metadata archival format

Metadata held alongside hetereogeneous data

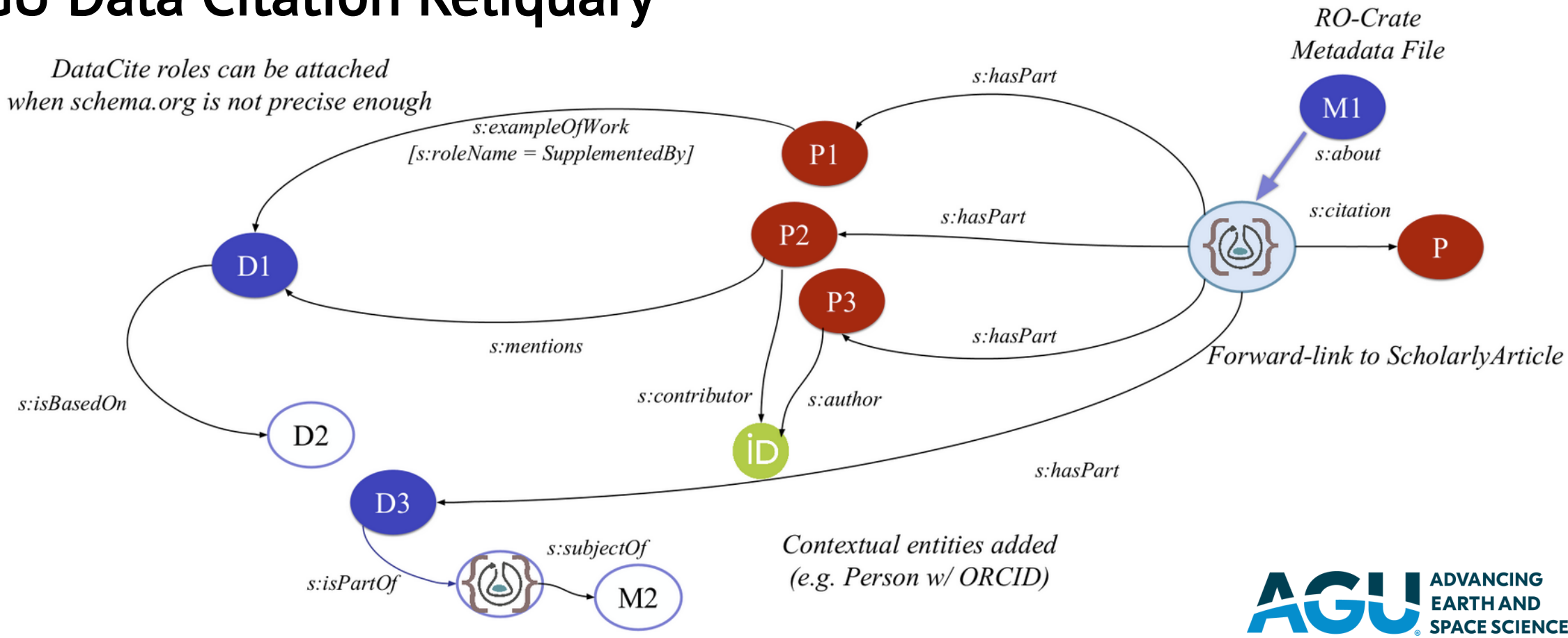
Exchange mechanism (import/export)

Avoid vendor lock-in

Oxford Common File Layout

Enabling large number of data citations: AGU Data Citation Reliquary

View is centric from particular RO-Crate
(Relations go mainly in one direction)
Uses schema.org vocabulary *s:*



D3 is also part of another RO-Crate, but is detailed in its own Metadata file

Credit: Paolo Manghi
AGU Data Citation Workshop
@soilandreyes

https://doi.org/10.5281/zenodo.4916734

RO-Crate in Earth Science

Earth Science RO-Crate profile (ongoing work)

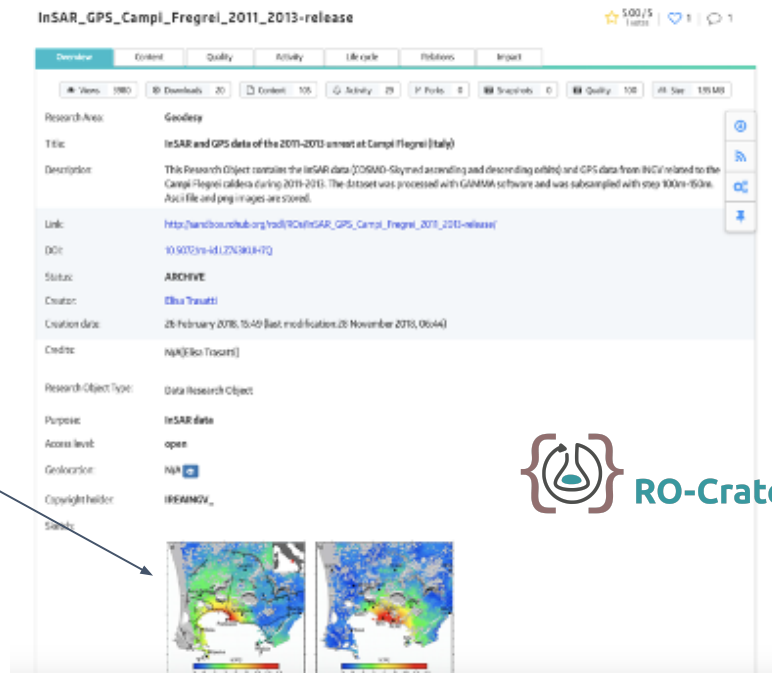
Data Cubes (**Collections and Products**) are very common in Earth Science research

We create the concept of **Data Cube Data Entity**

Relevant **metadata** for a Data Cube

- Simple **geolocation** (sch:contentLocation) and **temporal coverage** (sch:temporalCoverage)
- **Spatial coverage** (extent, resolution, crs)
- **Temporal coverage** (startTime, endTime, resolution-step)
- **Vertical coverage** (highestElevation, lowestElevation, resolution-step)
- Other (processing level, processor, instrument, platform, etc.)

http://www.rohub.org/rodetails/InSAR_GPS_Campi_Fregrei_2011_2013-release/



A valid RELIANCE *RO-Crate JSON-LD* graph MUST describe:
1. The RO-Crate Metadata File Descriptor
2. The Root Data Entity
3. One or more Data Cube Data Entities
4. Zero or more Data Entities
5. Zero or more Contextual Entities

Credit: Oscar Corcho, Carole Goble
https://doi.org/10.5281/zenodo.4913285

@soilandreyes

RO-Crate for workflow descriptions

MC_COVID19like_Assembly_Reads

Version 1

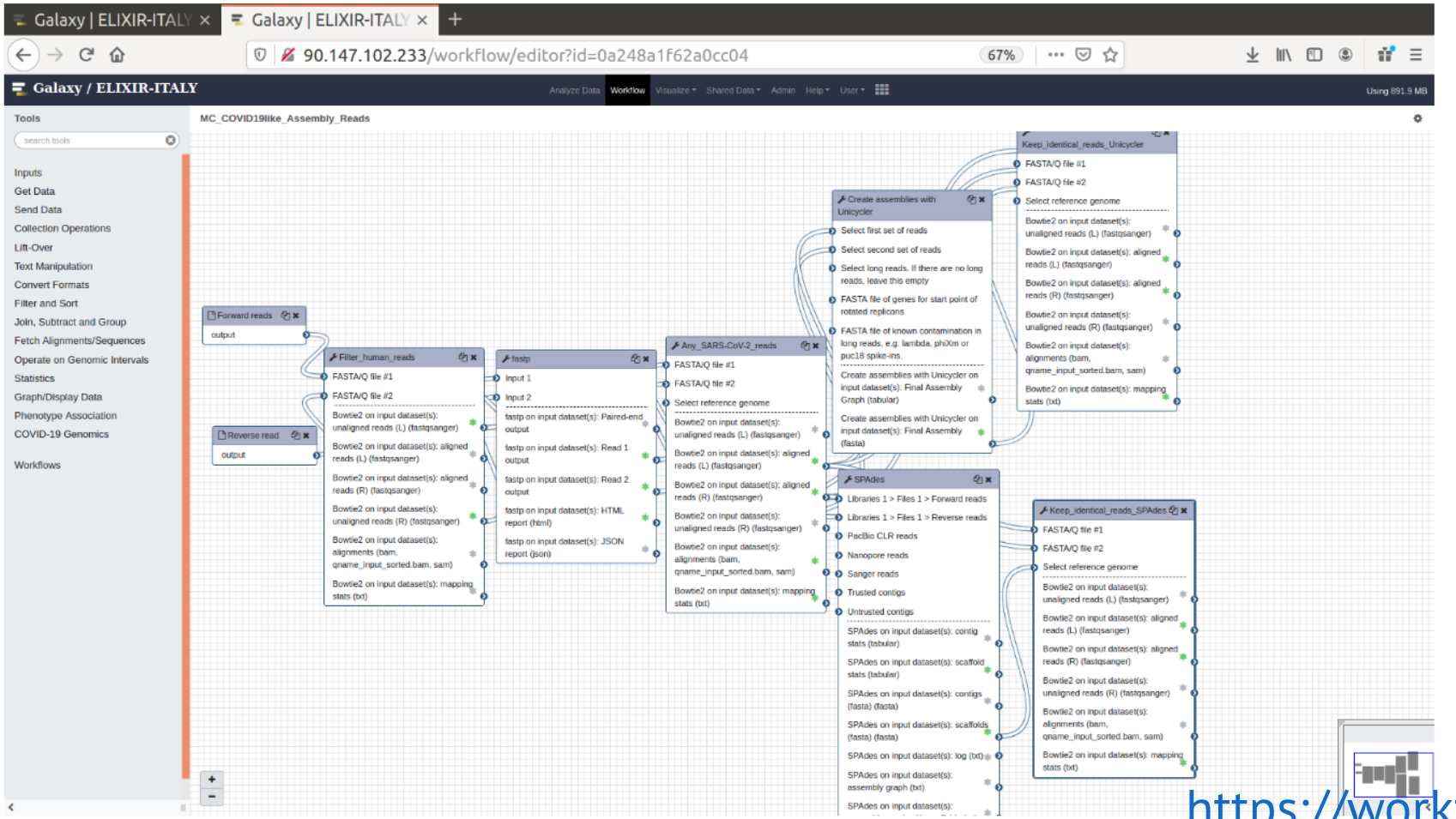
Work-In-progress

WorkflowHub

Visit source | Download RO Crate

This WF is based on the official Covid19-Galaxy assembly workflow as available from https://covid19.galaxyproject.org/genomics/2-assembly/ . It has been adapted to suit the needs of the analysis of metagenomics sequencing data. Prior to be submitted to INDSC databases, these data need to be cleaned from contaminant reads, including reads of possible human origin.

The assembly of the SARS-CoV-2 genome is performed using both the Unicycler and the SPAdes assemblers, similar to the original WV.



Creators and Submitter

Creators

Not specified

Submitter

Matteo Chiara

License

Apache Software License 2.0

Activity

Views: 78 **Downloads:** 2

Created: 4th Nov 2020 at 18:35
Last updated: 5th Nov 2020 at 07:42
Last used: 2nd Dec 2020 at 06:56

Tags

covid-19

Attributions

None

SEEK ID: https://workflowhub.eu/workflows/68?version=1

https://workflowhub.eu/workflows/68

Describing workflows with RO-Crate

Authors and contributors

ORCID

Life Monitor

Tests

Workflows

nextflow

Galaxy PROJECT

Snakemake

COMMON WORKFLOW LANGUAGE

Describe workflow

Bioschemas

RO-Crate

Registry

Containers

doi

WorkflowHub

https://about.workflowhub.eu/Workflow-RO-Crate/

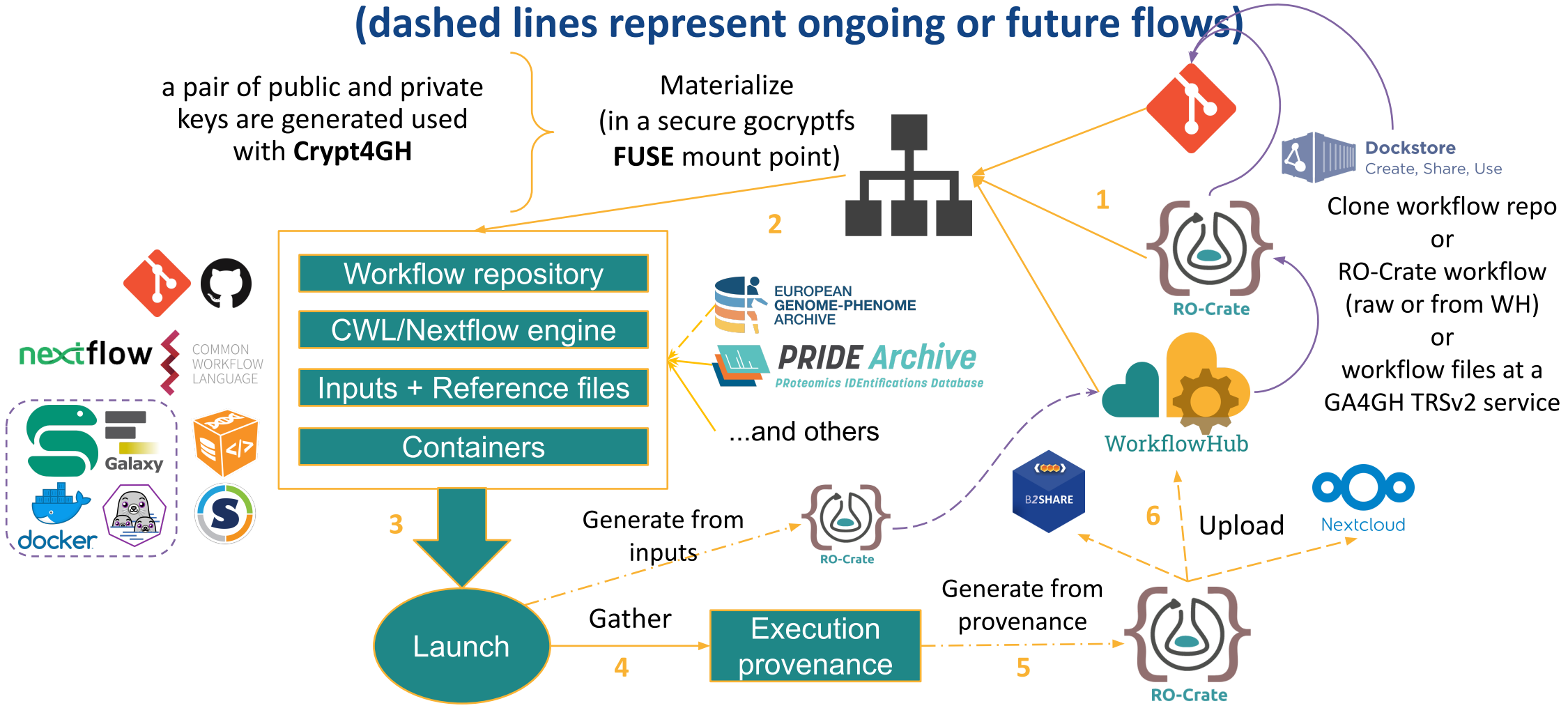
https://pypi.org/project/galaxy2cwl/

@soilandreyes

RO-Crate for workflow run provenane

Analysis life-cycle using WfExS

(dashed lines represent ongoing or future flows)



https://github.com/inab/WfExS-backend/

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Credit: José Mª Fernández, ELIXIR All Hands, 2021-06-11

Preserving a Workflow Run as RO-Crate?

- Workflow **language** & **version**
- Workflow **engine** & version (e.g. Toil)
- Workflow **definition**
- **Input** data (or pointers to such)
- **Parameters**? What can be implicit and explicit?
- Tool **Dependencies** to install (mostly implied by workflow?)
- **Container** platform requirement [e.g. Docker, Conda]
- **Operating system** requirement
- **Hardware** requirements (memory, CPU, GPU)
  - Equivalent of AWS **cloud instance** type sufficient?
- **Where to run**/submit (e.g. usegalaxy.eu)
- Explicit/resolved **container IDs**
- **Archive containers** from Docker Hub (protect against image expiration)
- …

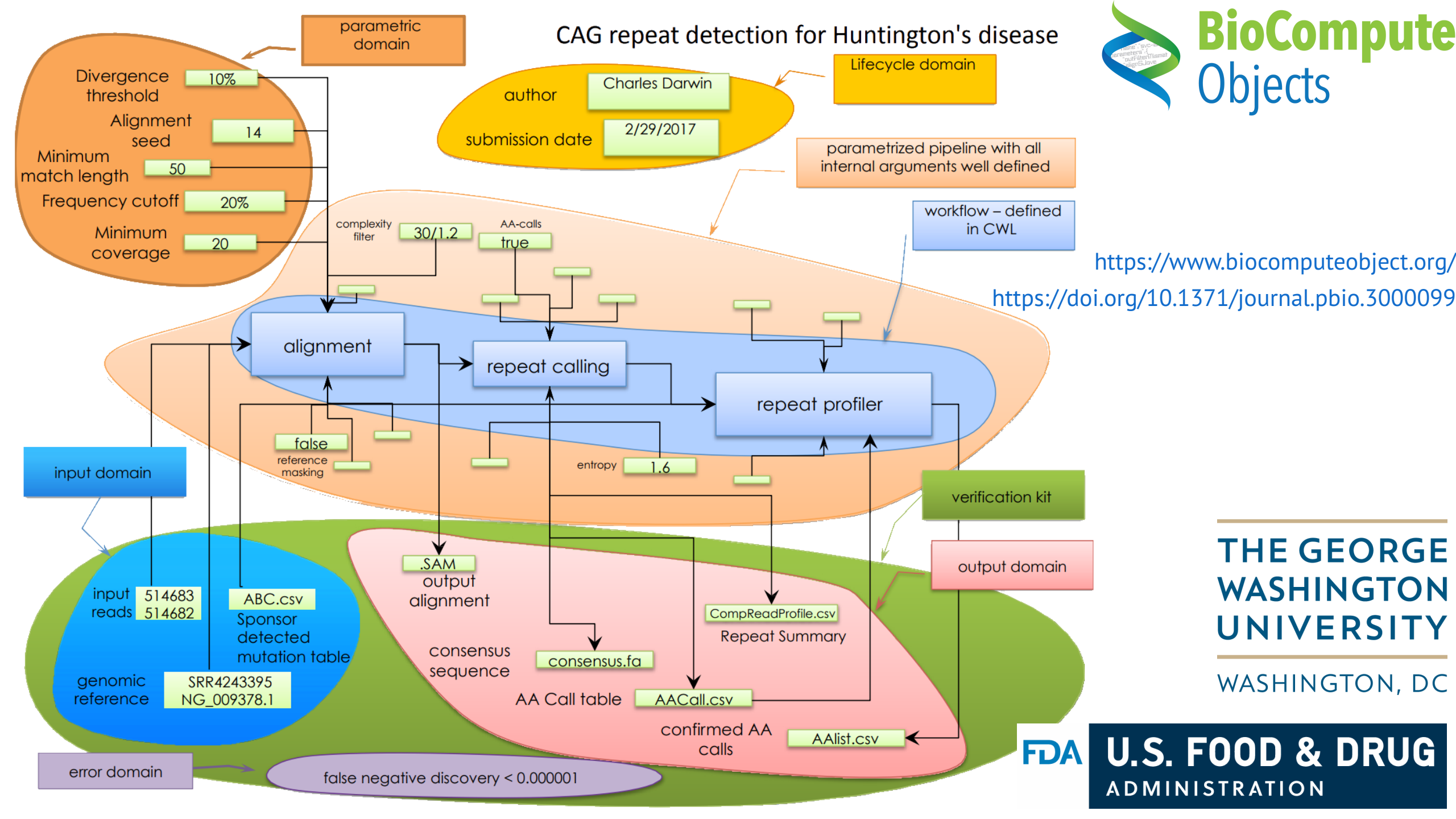
Join discussion in the
Workflow Hub Club community!
https://about.workflowhub.eu/

@soilandreyes

EOSC-*Life*

RO-Crate for regulatory sciences

@soilandreyes



BioCompute Objects

https://www.biocomputeobject.org/

https://doi.org/10.1371/journal.pbio.3000099

THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

FDA U.S. FOOD & DRUG ADMINISTRATION

JSON Schema

IEEE

IEEE2791-2020

Metadata

object_id : https://beta.portal.aws.biochemistry.gwu.edu/bco/BCO_00016916
spec_version : https://w3id.org/ieee/ieee-2791-schema/
etag : fea7e938e6bdf9a2cfcba7fa02f5a5fc3973dccb0b03a64319e1ee29966a5b6b

provenance_domain :
- embargo :
- created : 2020-08-04T23:50:56.016Z
- modified : 2020-08-04T23:50:56.016Z
- name : Human Healthy Bulk RNA-seq Expression (Bgee)
- version : v-1.0
- obsolete_after : 2020-04-22T23:57:00.000Z
- contributors :
  - contribution :
    - createdBy
  - name : Amanda Bell
  - email : amandab2140@gwu.edu
  - affiliation : GW HIVE-Lab
  - orcid : http://orcid.org/0000-0002-9920-565X
- license : Attribution 4.0 International CC BY 4.0

Provenance Domain

description_domain :
- keywords :
  - Gene Expression
  - Gene Expression Regulation
  - Tissue specificity
- xref :
  - namespace : ensembl
  - name : Ensembl Genome Browser
  - ids :
    - Ensembl gene ID
  - access_time : 2020-04-22T14:03:00.000Z
- platform :
  - OncoMX
- pipeline_steps :
  - step_number : 1
  - name : oncomx server
  - prerequisite :
    - uri :
  - description : Process data
  - input_list :

Description Domain

error_domain :
- empirical_error:
  - D168Y: percentage: 0.56, calls: 0.5615, STDEV.P: 0.00075
- algorithmic_error:
  - SCORE_threshold: 0.5, QUALITY: 25, COVERAGE: 5000

Error Domain

parametric_domain :
- param : grep
- value : -r
- step : 1

Parametric Domain

execution_domain :
- environment_variables :
  - key : EDITOR
  - value : vim
  - key : HOSTTYPE
  - value : x86_64-linux
- external_data_endpoints :
  - url : https://data.oncomx.org/ONCOMXDS000012
  - name : Human Healthy Bulk RNA-seq Expression (Bgee)
- script :
  - uri :
    - filename : make-dataset.py
    - uri : http://data.oncomx.org/ln2wwwdata/software/pipeline/integrator/make-dataset.py
    - access_time : 2020-04-22T14:28:00.000Z
- software_prerequisites :
  - uri :
    - filename : shell
    - uri : https://www.python.org/download/releases/2.7.5
    - access_time : 2020-04-22T14:30:00.000Z
  - name : Python
  - version : 2.7.5
- script_driver : Python

Execution Domain

io_domain :
- input_subdomain :
  - uri :
    - filename : Homo_sapiens_UBERON:0000066
    - uri : http://data.oncomx.org/ln2wwwdata/downloads/bgee/current/Homo_sapiens_UBERON:0000066_AFFYMETRIX_RNA_SEQ.tsv
    - access_time : 2020-04-22T20:44:00.000Z
- output_subdomain :
  - uri :
    - filename : human_normal_expression.csv
    - uri : https://data.oncomx.org/ONCOMXDS000012
    - access_time : 2020-04-22T20:50:00.000Z
  - mediatype : TEXT/CSV

IO Domain

extension_domain :
- dataset_categories :
  - category_value : Homo sapiens
  - category_name : species
  - category_value : normal
  - category_name : disease_status
- extension_schema : https://data.oncomx.org/ONCOMXDS000012

Extension Domain

usability_domain :
List of human taxid:9606 genes with healthy RNA-Seq and Affymetrix expression data in Bgee; additional documentation available at (https://github.com/BgeeDB/bgee_pipeline/tree/develop/pipeline/collaboration/oncoMX#information-about-the-files-generated-for-oncomx) Only the subset of RNA-Seq data are used to generate the expression profiles for healthy individuals for human used by OncoMX.

Usability Domain

BioCompute Objects

8 Top Level Domains	
Provenance Domain: Metadata describing the BCO	
Usability Domain: Free text field for researcher to explain the analysis and relevant details.	
Extension Domain: User-defined fields	
Description Domain: Steps of the analysis, external resources needed for the steps, and the relationship of I/O objects	
Execution Domain: Information about the environment in which the analysis was run	
Parametric Domain: Records any parameters that were changed from default values	
Input and Output Domain: A list of global input and output files	
Error Domain: Used for describing errors. Can include the limits of detectability, false positives, false negatives, statistical confidence of outcomes, and description of errors	
Required	**Optional**

docker

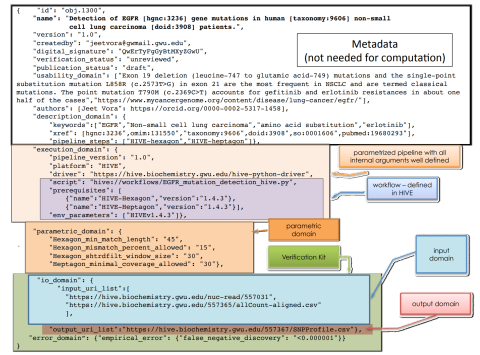
COMMON WORKFLOW LANGUAGE

identifiers.org

python™

BioCompute Objects

JSON



Alternate metadata views

Domain-specific explanation: BCO

General index: RO-Crate

```
{
  "@context": [
    "https://w3id.org/ro/crate/1.0/context",
    {
      "@vocab": "https://schema.org/"
    }
  ],
  "@graph": [
    {
      "@id": "ro-crate-metadata.json",
      "@type": "CreativeWork",
      "about": {
        "@id": "./"
      },
      "identifier": "ro-crate-metadata.json",
      "conformsTo": {
        "@id": "https://w3id.org/ro/crate/1.0"
      },
      "license": {
        "@id": "https://creativecommons.org/licenses/
      },
      "description": "Made with Describo: https://uts
    },
    {
      "@type": "Dataset",
      "author": {
        "@id": "https://orcid.org/0000-0001-9842-971
      },
```

RO-Crate

CSV

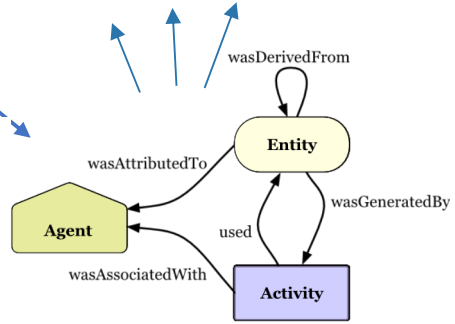
ORCID

ro-crate-metadata.json

BagIt

research object.org

JSON-LD



W3C® PROV

https://biocompute-objects.github.io/bco-ro-crate/

RO-Crate for computational tools

Software used to create files

To specify which software was used to create or update a file, the software application SHOULD be represented with an entity of type SoftwareApplication, with a version property, e.g. from `tool --version`.

For example:

```
{
      "@id": "https://www.imagemagick.org/",
      "@type": "SoftwareApplication",
      "url": "https://www.imagemagick.org/",
      "name": "ImageMagick",
      "version": "ImageMagick 6.9.7-4 Q16 x86_64 20170114 http://www.imagemagick.org"
}
```

The software SHOULD be associated with the File (or other data entities) it created as an instrument of a CreateAction, with the File referenced by a result property. Any input files SHOULD be referenced by the object property.

In the below example, an image with the `@id` of `pics/2017-06-11%2012.56.14.jpg` was transformed into an new image `pics/sepia_fence.jpg` using the *ImageMagick* software application as "instrument". Actions MAY have human-readable names, which MAY be machine generated for use at scale.

```
{
      "@id": "#Photo_Capture_1",
      "@type": "CreateAction",
      "agent": {
        "@id": "https://orcid.org/0000-0002-3545-944X"
      },
      "description": "Photo snapped on a photo walk on a misty day",
      "endTime": "2017-06-11T12:56:14+10:00",
      "instrument": [
        {
          "@id": "#EPL1"
```

https://www.researchobject.org/ro-crate/1.1/provenance.html#software-used-to-create-files

https://bioschemas.org/profiles/ComputationalTool/

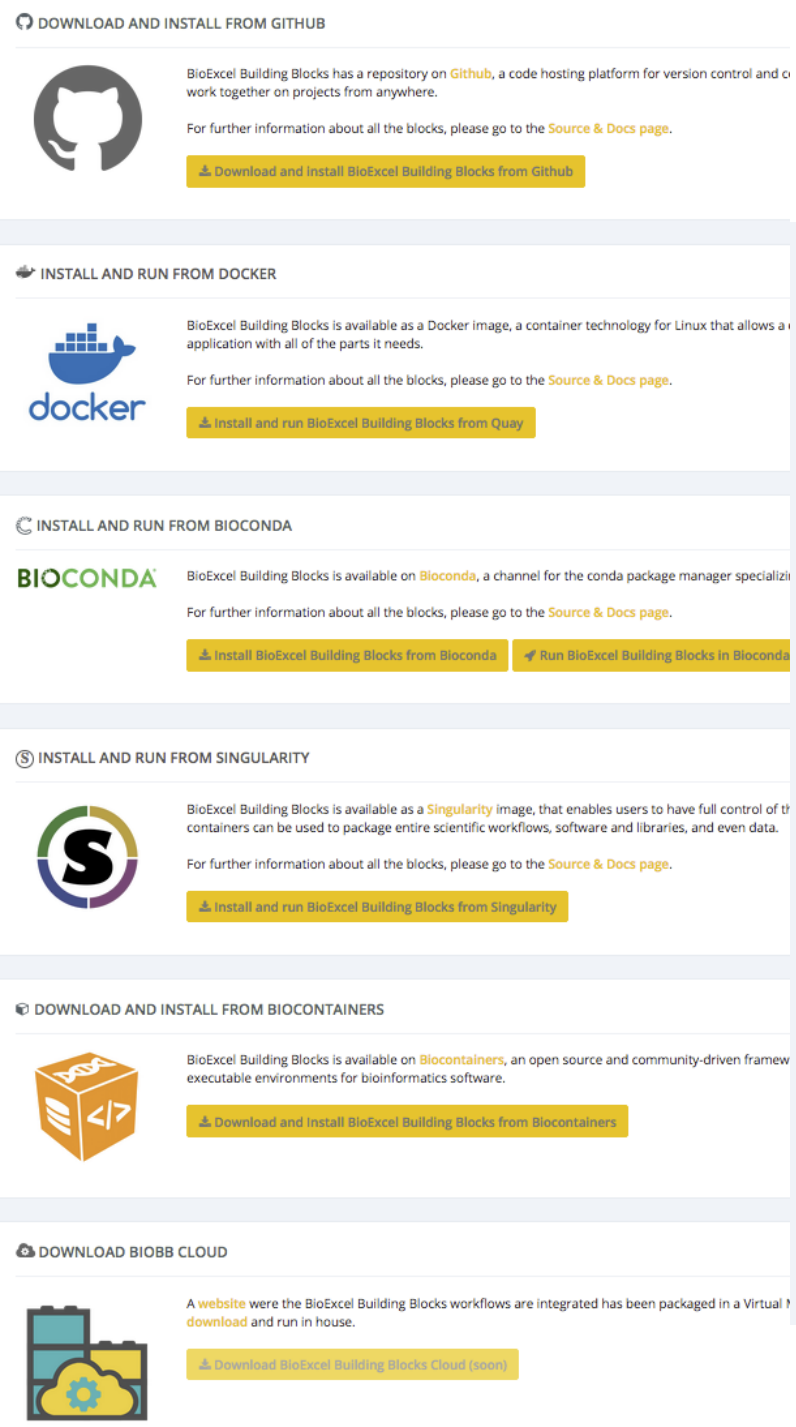
ındreyes

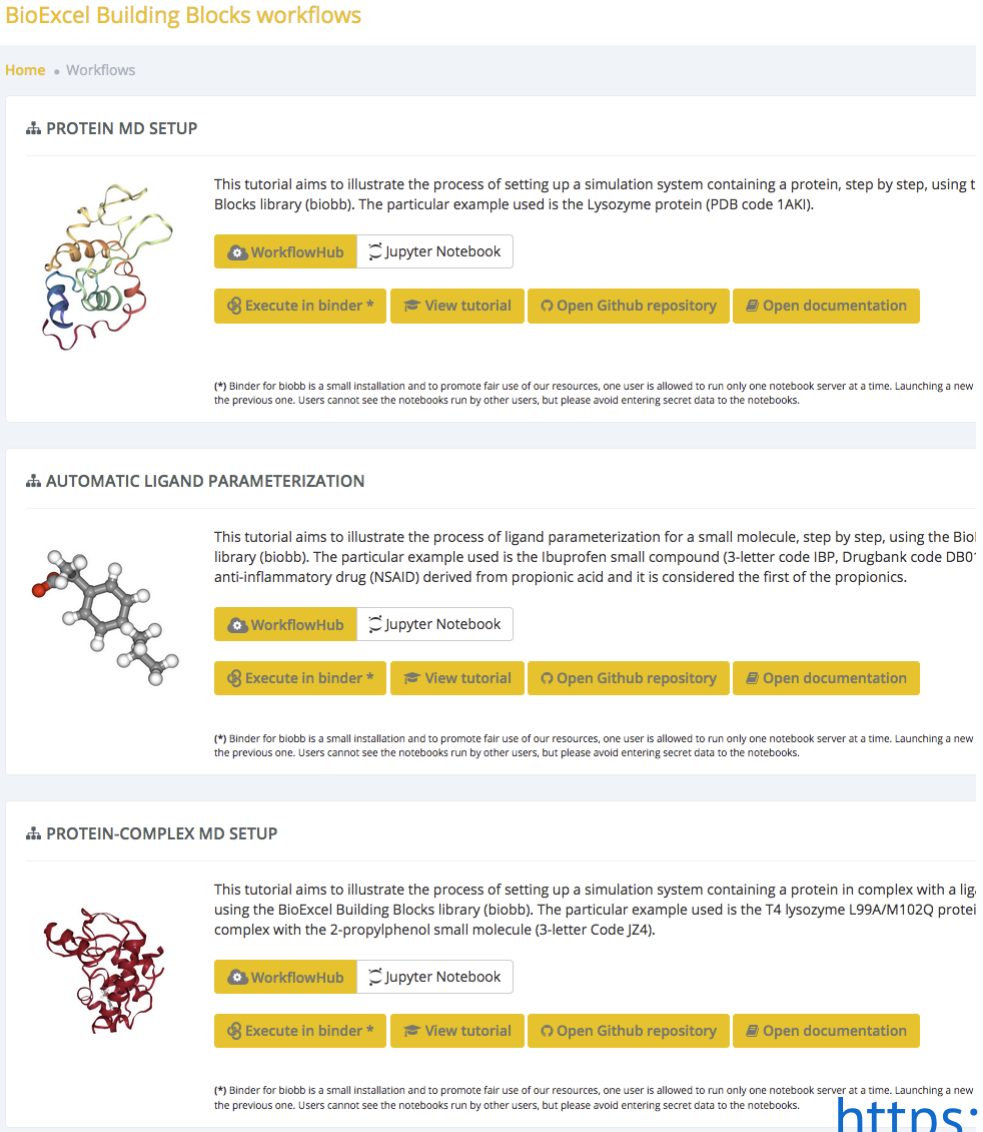
bioexcel
Center of Excellence for Computational Biomolecular Research

Use case:
HPC simulation Workflow using building blocks

@soilandreyes

Making Canonical Workflow Building Blocks interoperable across workflow languages





https://mmb.irbbarcelona.org/biobb/

@soilandreyes

https://doi.org/10.5281/zenodo.5542553

https://raw.githack.com/bioexcel/biobb_hpc_workflows/condapack/ro-crate-preview.html

https://stain.github.io/ro-crate-paper/

Accepted

https://stain.github.io/ro-crate-paper/

https://doi.org/10.5281/zenodo.5146228

arXiv:2108.06503

@soilandreyes

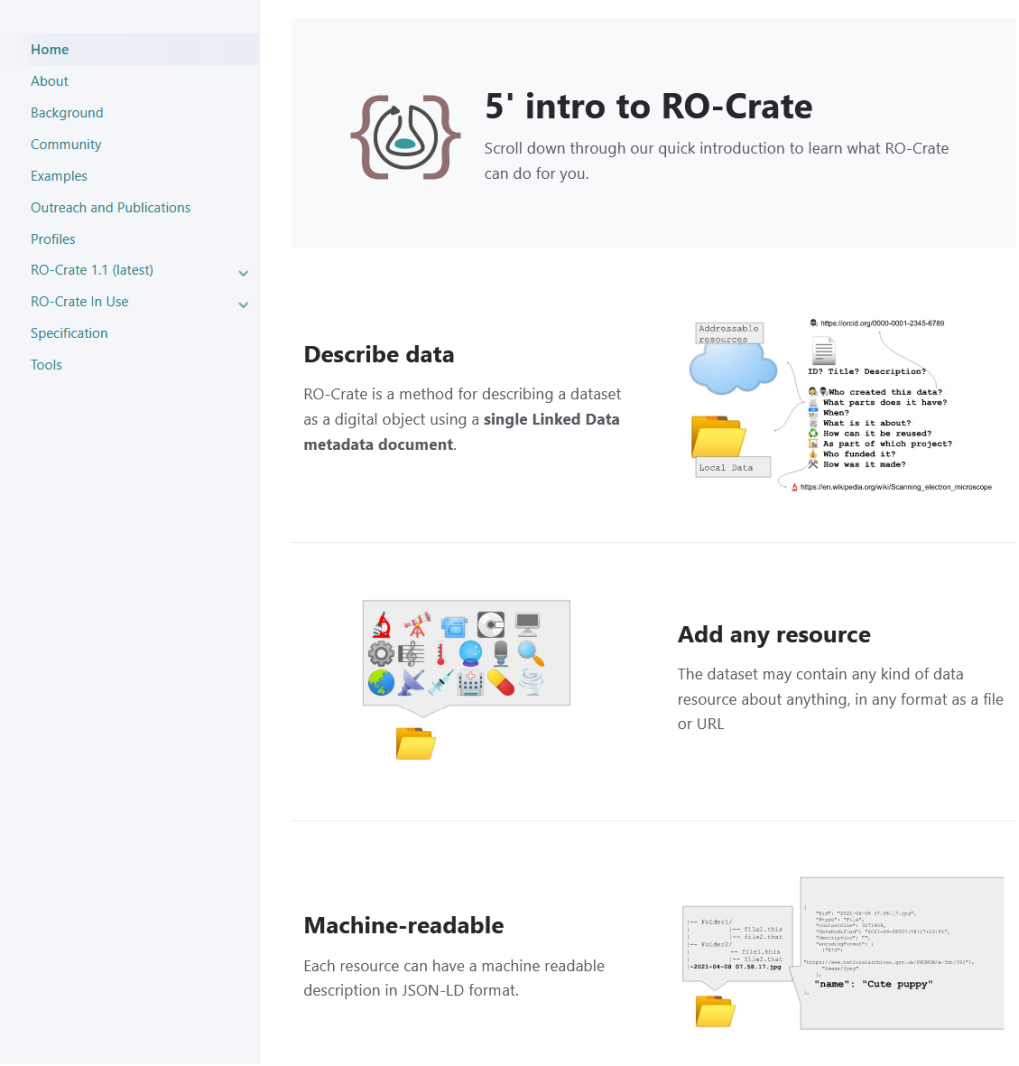
- Peter Sefton https://orcid.org/0000-0002-3545-944X (co-chair)
- Stian Soiland-Reyes https://orcid.org/0000-0001-9842-9718 (co-chair)
- Eoghan Ó Carragáin https://orcid.org/0000-0001-8131-2150 (emeritus chair)
- Oscar Corcho https://orcid.org/0000-0002-9260-0753
- Daniel Garijo https://orcid.org/0000-0003-0454-7145
- Raul Palma https://orcid.org/0000-0003-4289-4922
- Frederik Coppens https://orcid.org/0000-0001-6565-5145
- Carole Goble https://orcid.org/0000-0003-1219-2137
- José María Fernández https://orcid.org/0000-0002-4806-5140
- Kyle Chard https://orcid.org/0000-0002-7370-4805
- Jose Manuel Gomez-Perez https://orcid.org/0000-0002-5491-6431
- Michael R Crusoe https://orcid.org/0000-0002-2961-9670
- Ignacio Eguinoa https://orcid.org/0000-0002-6190-122X
- Nick Juty https://orcid.org/0000-0002-2036-8350
- Kristi Holmes https://orcid.org/0000-0001-8420-5254
- Jason A. Clark https://orcid.org/0000-0002-3588-6257
- Salvador Capella-Gutierrez https://orcid.org/0000-0002-0309-604X
- Alasdair J. G. Gray https://orcid.org/0000-0002-5711-4872
- Stuart Owen https://orcid.org/0000-0003-2130-0865
- Alan R Williams https://orcid.org/0000-0003-3156-2105
- Giacomo Tartari https://orcid.org/0000-0003-1130-2154
- Finn Bacall https://orcid.org/0000-0002-0048-3300
- Thomas Thelen https://orcid.org/0000-0002-1756-2128
- Hervé Ménager https://orcid.org/0000-0002-7552-1009
- Laura Rodríguez-Navas https://orcid.org/0000-0003-4929-1219
- Paul Walk https://orcid.org/0000-0003-1541-5631
- brandon whitehead https://orcid.org/0000-0002-0337-8610
- Mark Wilkinson https://orcid.org/0000-0001-6960-357X
- Paul Groth https://orcid.org/0000-0003-0183-6910
- Erich Bremer https://orcid.org/0000-0003-0223-1059
- LJ Garcia Castro https://orcid.org/0000-0003-3986-0510
- Karl Sebby https://orcid.org/0000-0001-6022-9825
- Alexander Kanitz https://orcid.org/0000-0002-3468-0652
- Ana Trisovic https://orcid.org/0000-0003-1991-0533
- Gavin Kennedy https://orcid.org/0000-0003-3910-0474
- Mark Graves https://orcid.org/0000-0003-3486-8193
- Jasper Koehorst https://orcid.org/0000-0001-8172-8981
- Simone Leo https://orcid.org/0000-0001-8271-5429
- Marc Portier https://orcid.org/0000-0002-9648-6484
- Paul Brack https://orcid.org/0000-0002-5432-2748
- Milan Ojsteršek https://orcid.org/0000-0003-1743-8300
- Bert Droesbeke https://orcid.org/0000-0003-0522-5674
- Chenxu Niu https://orcid.org/0000-0002-2142-1731
- Kosuke Tanabe https://orcid.org/0000-0002-9986-7223
- Tomasz Miksa https://orcid.org/0000-0002-4929-7875
- Marco La Rosa https://orcid.org/0000-0001-5383-6993
- Cedric Decruw https://github.com/cedricdcc
- Andreas Czerniak https://orcid.org/0000-0003-3883-4169
- Jeremy Jay https://orcid.org/0000-0002-5761-7533
- Sergio Serra https://orcid.org/0000-0002-0792-8157
- Ronald Siebes https://orcid.org/0000-0001-8772-7904
- Shaun de Witt https://orcid.org/0000-0003-4196-3658
- Shady El Damaty https://orcid.org/0000-0002-2318-4477
- Douglas Lowe https://orcid.org/0000-0002-1248-3594
- Sergio Serra https://orcid.org/0000-0002-0792-8157
- Xuanqi Li https://orcid.org/0000-0003-1498-6205

RO-Crate Community is open for anyone to join us!

https://www.researchobject.org/ro-crate/community

More documentation

- 5 minute intro
- Tutorial / training material (Carpentries-style)
- Showcases of use



https://www.researchobject.org/packaging_data_with_ro-crate/

@soilandreyes

Future directions

Ongoing:

- CS3mesh4EOSC (Describo Online w/ cloud storage)
- EOSC-Life (Life Science workflows)
- BY-COVID (COVID-19)
- SYNTHESYS+/DiSSCO (FAIR Digital Objects to digitize museum collections)
- GA4GH & ELIXIR Cloud (workflow execution)

Next round (proposals and collaborations):

- Workflows as FAIR Digital Objects
- Digital Twins in Biodiversity
- Minimal metadata standards for Semantic Interoperability
- Cross-repository integrations