

# FAIR Computational Workflows

Sarah Cohen-Boulakia, Université Paris-Saclay

Joint work with

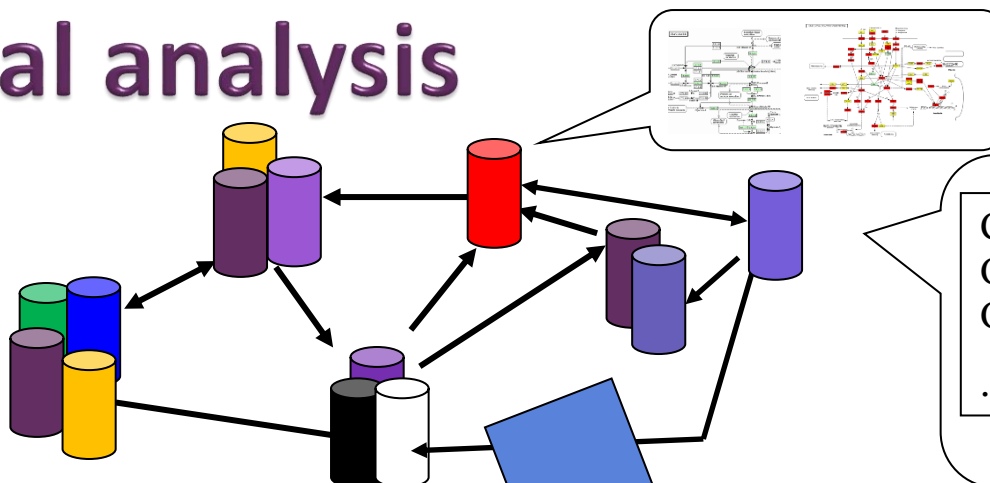
Carole Goble, Stian Soiland-Reyes,  
Daniel Garijo, Yolanda Gil,  
Michael R. Crusoe,  
Kristian Peters & Daniel Schober

# Biological analysis

## Public sources

- Distributed
- Heterogeneous
- Network

> 1,500 (NAR)



```
CCCTTTCCCGTGT
G TCCCGTCTCCG
G T
C T
..
TGCCGTGTGGC
TAAATGTCTGTG
GTCTGTGC...
```

Tools

Scripts



Python

JAVA

Web services

Analysis pipelines

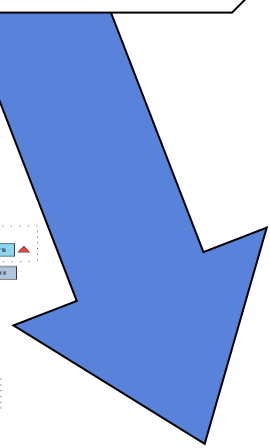
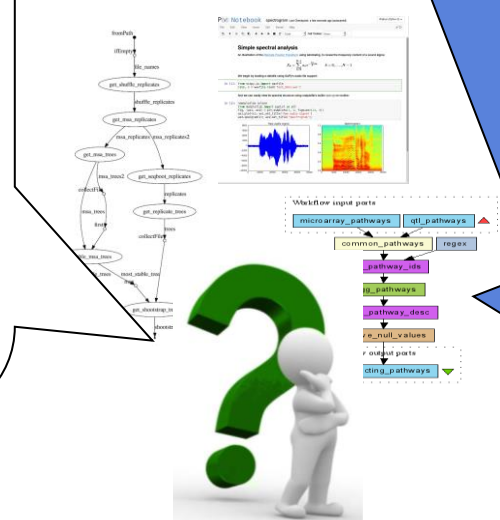
## Tools - Scripts

- Distributed
  - Heterogeneous
- > 17,300 (bio.tools)

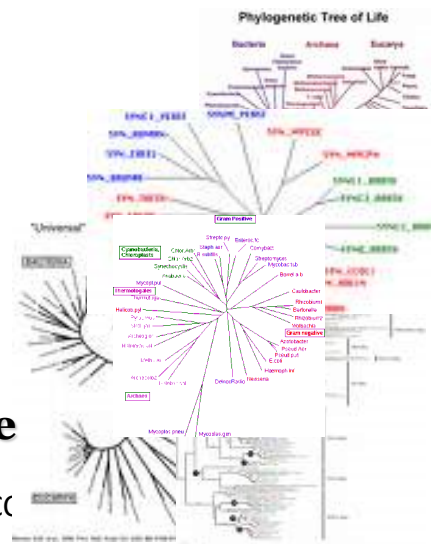
Need to be able to understand/interpret the results...

*How the data have been obtained? Which tools? Which data?*

Need to share my result with colleagues.



Workspace



# Variety of means to perform data analysis

- ▶ From scripts ... to Notebooks



- ▶ **Workflow Management Systems**

- *coarse-grained*: chaining locally hosted or distributed tools
- *fine-grained*: optimizing computational resources (distributed infrastructure, HPC, cloud-based container orchestration...)



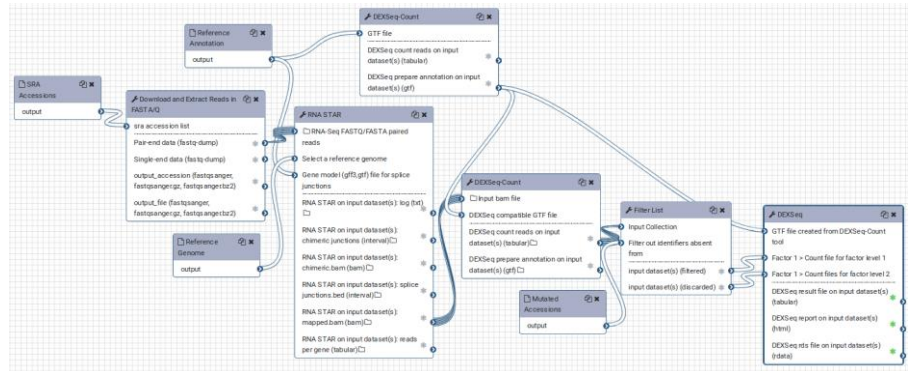
- ▶ **Possible features of WfMS**

- **User interactions**: APIs vs scripting vs GUI
- **Resource scalability**: optim, concurrency and parallelisation
- **Portability management** : dependencies on the infra
- **Secure execution**: monitoring and fault handling
- **Tracking**: process logging and data provenance tracking
- **Data handling**: secure access, movement, ref management

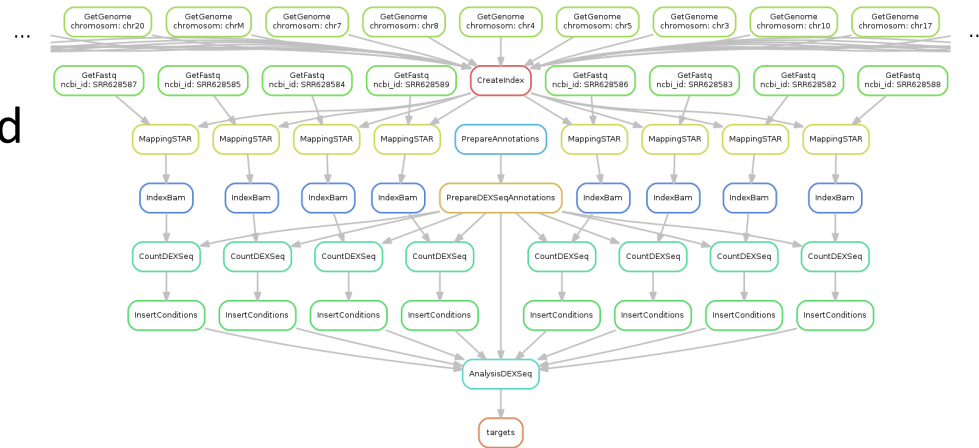


# Computational workflows

- ▶ Separation of the workflow specification from its execution
- ▶ Are they computational workflow?
  - Workflows from most **WfMS** ✓
  - **Notebooks** when the dataflow is explicit (cells) ✓
  - **Scripts** usually interleave data and computational processes
    - **YesWorkflow** provides means to annotate scripts



Precise description of a procedure: multi-step process coordinated by input/output data relationships (data types)



Execution of a computational process (running a code, invocation of a service...). Data is consumed and produced by each step.

# Outline

Computational workflows

FAIR data **for and from** workflows

FAIR criteria for **workflows as digital objects**

Conclusion

# FAIR Principles

## Findable

- F1.** (Meta)data are assigned a globally unique and persistent identifier
- F2.** Data are described with rich metadata (defined by R1 below)
- F3.** Metadata clearly and explicitly include the identifier of the data they describe
- F4.** (Meta)data are registered or indexed in a searchable resource

## Accessible

- A1.** (Meta)data are retrievable by their id using a standardised communications protocol
  - A1.1** The protocol is open, free, and universally implementable
  - A1.2** The protocol allows for an authentication and authorisation procedure, where necessary
- A2.** Metadata are accessible, even when the data are no longer available

## Interoperable

- I1.** (Meta)data use a formal, accessible, shared, and broadly applicable language for KR
- I2.** (Meta)data use vocabularies that follow FAIR principles
- I3.** (Meta)data include qualified references to other (meta)data

## Reusable

- R1.** (Meta)data are richly described with a plurality of accurate and relevant attributes
  - R1.1.** (Meta)data are released with a clear and accessible data usage license
  - R1.2.** (Meta)data are associated with detailed provenance
  - R1.3.** (Meta)data meet domain-relevant community standards

# FAIR data for and from workflows

## ▶ FAIR data...

- **Open ontologies, vocabularies and services** for data interoperability and identification resolution

- **MIAPPE/Breeding API** (BrAPI): interface for data exchange
- **EDAM ontology**: input / output of tools executed

→ *Data interoperability (I1, I2, I3) and identifier resolution (F1, A1)*

## ▶ ... for WfMS, allowing to make informed choices

- On the specification phase: suggesting tools,...
- On the execution phase: validating data type,...

## ▶ Combination of FAIR data and FAIR tools: **FAIR e-infrastructure**



## ▶ Well-designed workflow management systems can automate the production of FAIR data

- Metadata descriptions of data products
- **Deposition of data in searchable resources**

→ *(F2, I2, I3, R1.3) and (F4)*

# Challenges in workflow execution

## ► Identifiers (F1, F3, A1)

- Propagation of ids through the workflow
- Tracking data attribution and the minting of ids for numerous intermediate results
  - **Minids** : light-weight id to unambiguous name, identify and reference research data products
- Wf need to move data ref through their engines (not the data itself)

## ► Licensing (R1.1)

- **Combining licenses** impact licensing the workflow or its data products

## ► Data access (A1.1, A1.2)

- workflow constituents require **harmonized Authentication and Authorization Infrastructure (AAI)** propagation through the different tasks, hosted by different service providers using different operating systems





# Additional challenges

## ▶ Workflow Provenance


- WfMS provides **documentation** of how the data has been generated (**R1.2**)
  - Standardisation efforts  model and ontology (**I1, I2**)

**But...** Provenance standards have to be fully embraced by WfMS

- Lack of provenance processing tools
- Automated provenance collection can be too fine grained and too detailed

## ▶ Steps in **coarse-grained workflows** may be wrapped applications

- Sub-workflows and (**not tracked**) steps within
- Data resources and tools may **not report basic metadata** (version, licence) in a standardised, machine interpretable way

→  Bioschemas : metadata marked-up in resources in a lightweight way

## ▶ **unFAIR** service provision

- The **components change their interfaces** without notice, breaking workflows

# Outline

Computational workflows

FAIR data **for and from** workflows

FAIR criteria for **workflows as digital objects**

Conclusion

# FAIR criteria for workflows – Wf Repositories

▶ FAIR criteria have been **envisioned for data**

▶ **Workflow registries dedicated** to WfMs

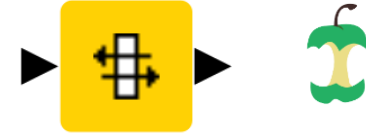
◦ **KNIMEHub**, **nf-core** (Nextflow), snakemake-wrappers

◦ findability-accessibility (F4), description/metadata workflows (F2) and may provide persistent, unique ids (F1)

◦ Access is baked into the workflow systems (A1)

→ Accessibility = wf should be archived and cited using citation metadata

• **schema.org** mark-up used by **Datacite** (+ *tool&wf terms needed!*)



▶ **myExperiment**

◦ WfMS **agnostic** repository, **pioneering**: workflow finding/sharing/publishing

◦ laid the foundations for **workflow-based Research Objects**

▶ **WorkflowHub** (EOSC Life)



◦ **CWL** standards

◦ **Research Objects federated** (RO-Crate)

◦ Registries for tools (**bio.tools**) and containers (**Biocontainers**)

# FAIR criteria for workflows – Wf description

Attempts to standardise workflow descriptions in order to aid discoverability (F1) and enable interoperability (I1)

- ▶ The Interoperable Workflow Intermediate Representation (**IWIR**)
  - common bridge for translating fine-grain workflows in different languages, independent of the underlying distributed computing infra
- ▶ The **Workflow Description Language** and the **Common Workflow Language** are recent community efforts to describe workflows
  - CWL standards describe workflows+tool interfaces making them portable
    - scalable across a variety of software and hardware environments
    - runnable by other CWLcompliant engines



***As descriptions of processes workflows inherit properties of FAIR data, but as executable processes they inherit properties of software!***

# Challenges for FAIR workflows as processes

## Structure and Forms

- ▶ **Structure** : Workflows are often inherently **composite**
  - **Nested workflows**: *sub-workflows* executed as part of complex workflows
  - The distinction between a workflow and its *component steps* is blurred
- FAIR can be applied simultaneously on **multiple levels**
  - Findable composite workflows = findable involved tools and data types
  - FAIR on the components – metadata, licensing, ... – propagate to the wf level
    - **may be incompatible**
  - Identify, cite ... **composite, multi-authored objects** is an open question
- ▶ **Forms - FAIR workflow: what do we mean?**
  - a CWL **specification** with test or exemplar data
  - an **implementation** of that design in a WfMS
  - an **instantiation of that implementation** ready to run with input data, parameters set, computational services spun up
  - a **run result** with intermediate/final data products and provenance logs

**Workflow-centric Research Objects** attempt to create a metadata framework to capture each form, but **each may have different FAIR criteria**

# Challenges for FAIR workflows as processes

## Versioning & Executability

### ▶ Workflows are living artefacts

- **Workflow evolution** = a form of provenance (R1.2)



Workflows can be recycled, repurposed: cloned, forked, merged ...changed

- **nf-core**: collab dev env (github) natively versioning + **testing and validation**

FAIR for workflow must address **versioning and “fixivity”**: snapshot a workflow and its dependencies to fix its reproducible state + associate a persistent id

### ▶ Workflows are executable objects




- Interoperable + reusable = **portable**, encapsulating all runtime dependencies
  - Container-based virtualisation sol + platform indep software packaging/distribution

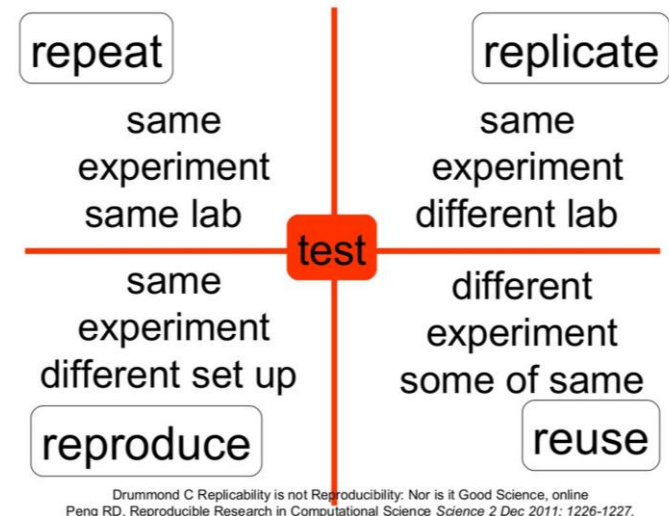
**But...** workflows and the software tools used are **time limited objects** whose **active lifespan is dependent** on that of their **components, WfMS, scientific relevance**

→CWL

# Challenges for FAIR workflows as processes

## Reuse

- ▶ Depends on the purpose of the reuse
- ▶ R1 is fostered by robust software practices
  - **Testing** workflow, modules, software tools
  - Interop = workflow replication on platforms
    - **OpenBench** 
- ▶ **Validation of parameters** to preclude workflow failure and faulty/unsafe results
  - The formulation of parameters must be FAIR
    - Doc of their purpose and range definitions
      - The **BioCompute Object specification**: representation and validation of parameters for reusable computational pipelines (precision medicine)



# Conclusion

- ▶ Workflows capture complex methods
  - **FAIR properties needed** to be published, finable, accessed, cited, reused...
- ▶ FAIR principles for data and for software are applicable but **need to be extended to capture the processual nature of workflows**
  - **Appropriate FAIR principles for software**, incorporating **best practices** for maintainability, maturity and reproducibility
  - Individual parts, forms, versions and execution environments of a workflow need to be FAIR and their **combination** too : **complex interdependencies to be covered** by additional FAIR metrics
- ▶ FAIRification of workflows pave the way for trustable data with the added value of being ready for exploitation by third parties





& Daniel Schober



# Thanks!

