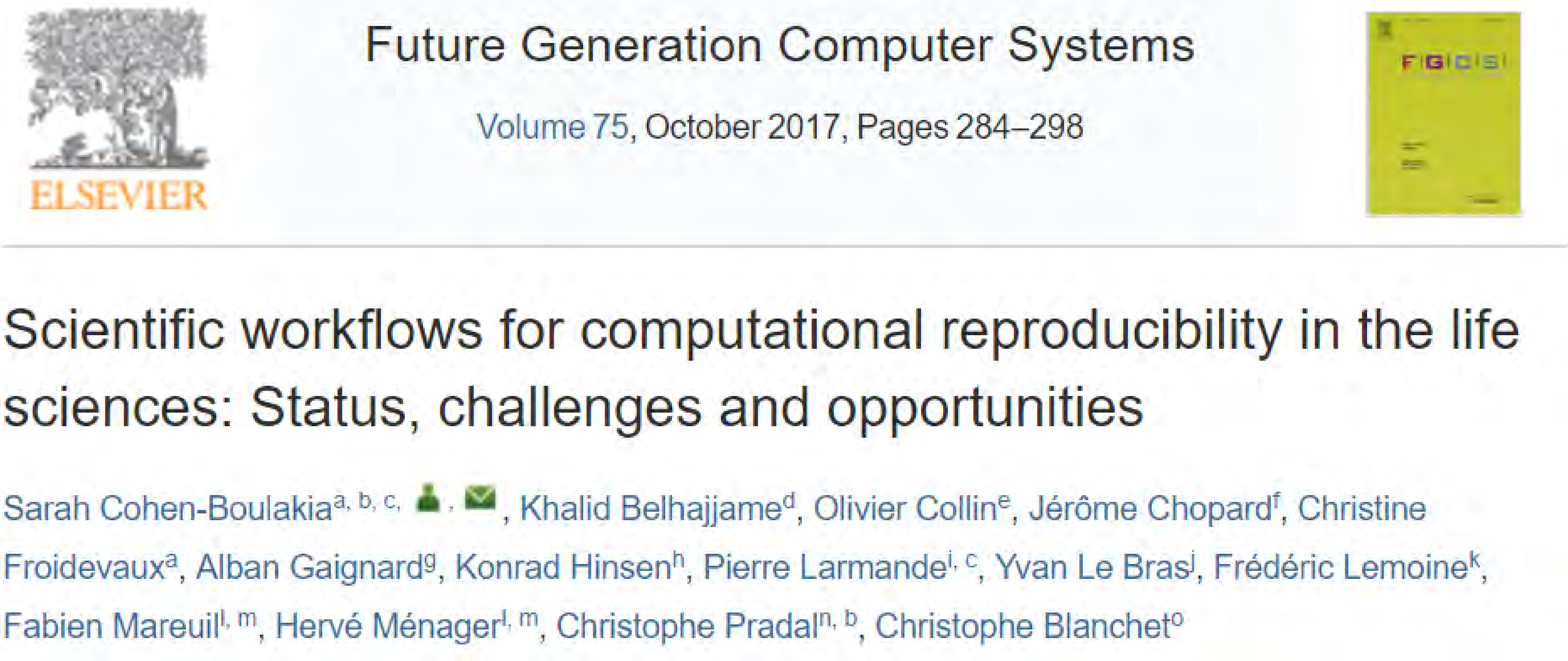


Future Generation Computer Systems

Volume 75, October 2017, Pages 284–298

Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities

Sarah Cohen-Boulakia[a, b, c], Khalid Belhajjame[d], Olivier Collin[e], Jérôme Chopard[f], Christine Froidevaux[a], Alban Gaignard[g], Konrad Hinsen[h], Pierre Larmande[i, c], Yvan Le Bras[j], Frédéric Lemoine[k], Fabien Mareuil[l, m], Hervé Ménager[l, m], Christophe Pradal[n, b], Christophe Blanchet[o]

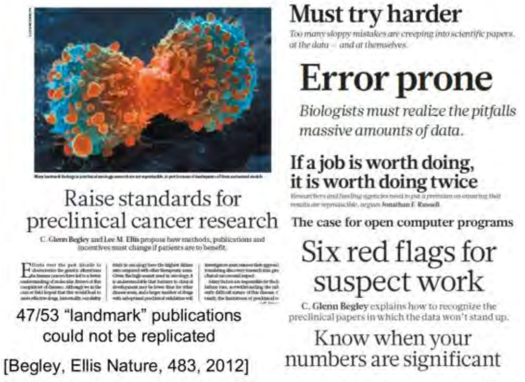
Another MaDICS success story!

Sarah Cohen-Boulakia

Université Paris-Sud, Laboratoire de Recherche en Informatique
CNRS UMR 8623, Université Paris-Saclay, Orsay, France

Context, Challenges

- *Computational reproducibility*
- Increasing number of irreproducible scientific results
  - Even published in high IF venues
  - Not (always) deliberately
- Various scientific domains
  - Consequences may be huge (preclinical studies…)
- Major challenge
  - The cost of irreproducible preclinical studies have been evaluated to >$10 Billions per year (USA)
- Becoming mandatory
  - NSF projects, editors…



Raise standards for preclinical cancer research

C. Glenn Begley and Lee M. Ellis propose how methods, publications and incentives must change if patients are to benefit.

47/53 “landmark” publications could not be replicated

[Begley, Ellis Nature, 483, 2012]

Must try harder

Too many sloppy mistakes are creeping into scientific papers. … at the data — and at themselves.

Error prone

Biologists must realize the pitfalls … massive amounts of data.

If a job is worth doing, it is worth doing twice

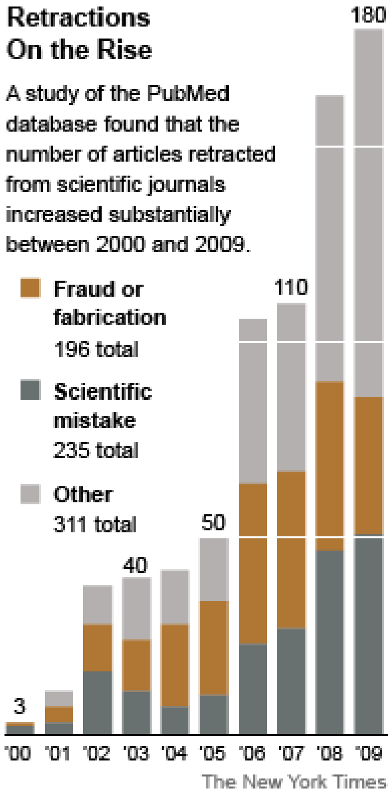
Researchers and funding agencies need to put a premium on ensuring that results are reproducible, argues Jonathan F. Russell

The case for open computer programs

Six red flags for suspect work

C. Glenn Begley explains how to recognize the preclinical papers in which the data won't stand up.

Know when your numbers are significant



Retractions On the Rise

A study of the PubMed database found that the number of articles retracted from scientific journals increased substantially between 2000 and 2009.

The New York Times

Aims of the action

- **Concepts, Needs/solutions**
  - Which *levels* of reproducibility can we consider?
  - Which are the solutions (methods and tools) currently available for *reproducibility*?
- **Opportunities, challenges**
  - What is missing?
  - Which are the *research* (vs technical) *open issues*?
- **Evaluation of solutions** based on **practice and state-of-the-art**
  - Experience of developers in using solutions in real contexts
  - ReproHackathon

    → Real use cases from the Bioinformatics Domain

Biological Data Analysis

- **From Data to Knowledge**
  - Data

    Distributed, **Heterogeneous**
  - Tools

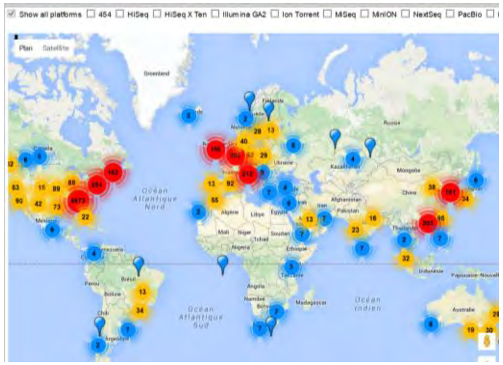
    Different kinds, various parameters
  - Analysis pipelines (*workflows*)

    Complex
- **Use cases**
  - NGS (cancer), Plant Phenotyping

    Big data sets
  - European Research Infrastructure

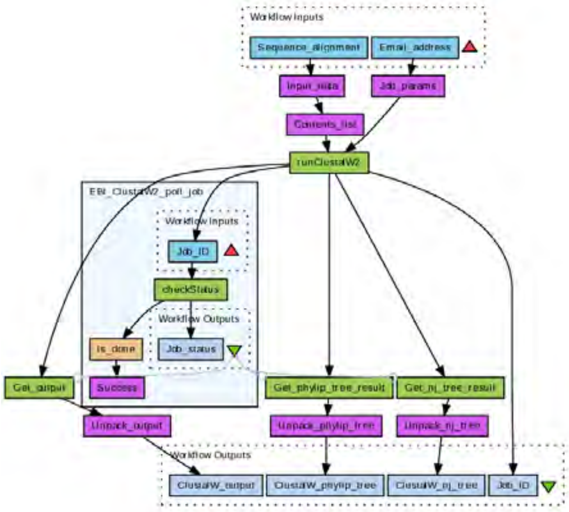
    21 countries, 180 partners

    ➔ Analyses with scientific workflow systems

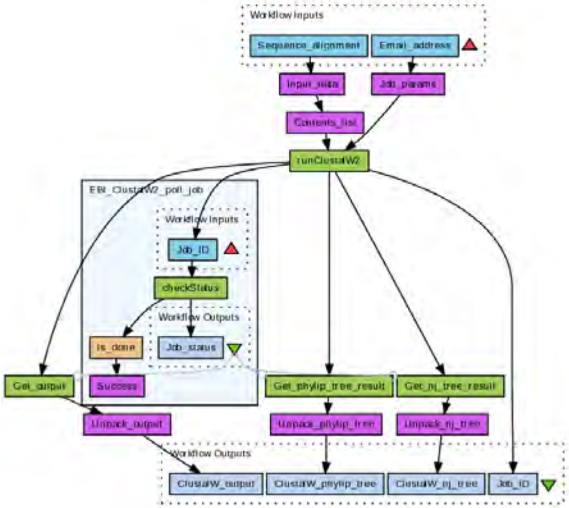

☑ Show all platforms □ 454 □ HiSeq □ HiSeq X Ten □ Illumina GA2 □ Ion Torrent □ MiSeq □ MinION □ NextSeq □ PacBio □
Plan
Satellite
Océan Atlantique Nord
Océan Atlantique Sud
Océan Indien

Scientific workflow systems

- Numerous systems: Galaxy, VisTrails, Taverna, NextFlow, OpenAlea …
- Specification vs Executions
  - Specification
    - Tools to be called, in which order
    - Workflow and components can be annotated and stored into repositories
  - Execution
    - The specification run with input dataset + parameter setting
    - Tracking, logging data produced and consumed



Scientific workflow systems



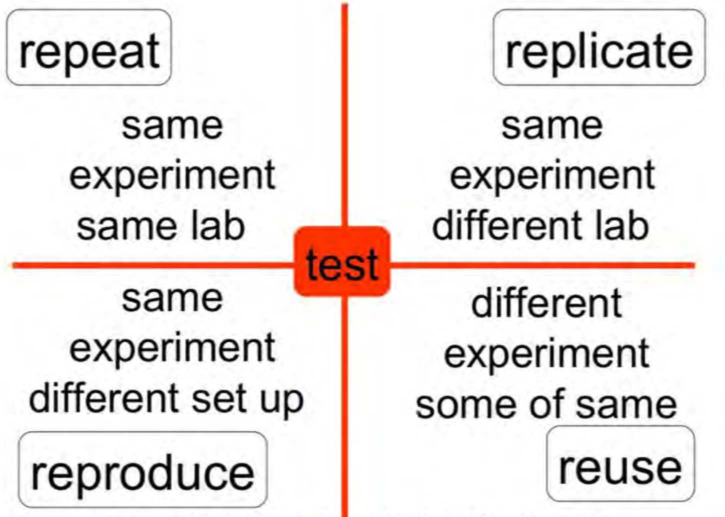
- Numerous systems: Galaxy, VisTrails, Taverna, NextFlow, OpenAlea …
- Specification vs Executions
  - Specification
    - Tools to be called, in which order
    - Workflow and components can be annotated and stored into repositories
  - Execution
    - The specification run with input dataset + parameter setting
    - Tracking, logging data produced and consumed

Which reproducibility levels when using workflow systems?
Which features for a *reproducibility-friendly* workflow system?

Outline

- Context
- Levels of reproducibility in scientific workflow systems
- Reproducibility-friendly features
- Open challenges

A continuum of possibilities



repeat

same experiment same lab

replicate

same experiment different lab

test

same experiment different set up

reproduce

different experiment some of same

reuse

Drummond C Replicability is not Reproducibility: Nor is it Good Science, online
Peng RD, Reproducible Research in Computational Science *Science 2 Dec 2011: 1226-1227.*

3 ingredients

Workflows Specification

Chained Tools

Workflow Execution

Input data and parameters

Environment

OS/librairies installed…

- Repeat
  - *Redo*: exact same context
  - Same workflow, execution setting, environement
  - Identical *output*

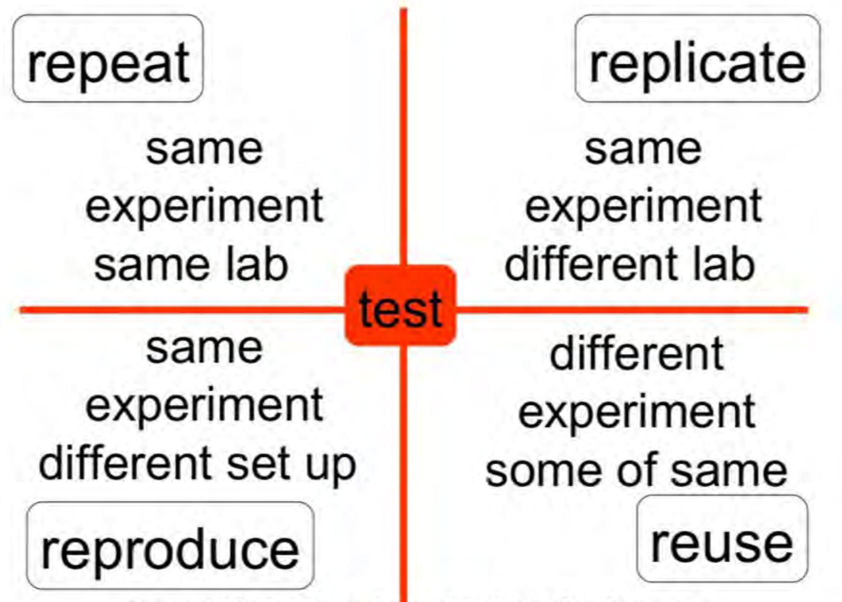
  →Aim = proof for reviewers ☺

- Replicate
  - Variation allowed in the workflows, execution setting, environement
  - Similar *output*

  → Aim = robustness

A continuum of possibilities

- Reproduce
  - Same *scientific result*
  - But the means used may be changed
  - Different workflows, execution setting, environment
  - Different output but in accordance with the result


repeat
same experiment same lab
replicate
same experiment different lab
test
same experiment different set up
reproduce
different experiment some of same
reuse

Drummond C Replicability is not Reproducibility: Nor is it Good Science, online
Peng RD, Reproducible Research in Computational Science *Science 2 Dec 2011: 1226-1227.*

- Reuse
  - Different scientific result
  - Use of tools/… designed in another context

Outline

- Context
- Levels of reproducibility
- Reproducibility-friendly features
- Open challenges

Reproducibility-friendly features in scientific workflows

5 Systems: Galaxy, VisTrails, Taverna, OpenAlea, NextFlow

Workflow specification

Language (XML, Python…) → repeat … reuse

Interoperability (CWL…) → replicate … reuse

Description of steps

- Remote services → repeat
- Command line → repeat … reuse
- Access to source code → replicate

Modularity (nested workflows?) → reuse

Annotation (tags, ontologies…) → reuse

Execution

Language and standard (PROV…,) → repeat … reuse

Presentation (interactivity with the results/provenance, notebooks) → replicate … reuse

Annotations → reuse

Reproducibility-friendly features in scientific workflows (cont.)

Environment (companion tools)

Ability to run workflows within a given environment → repeat (… reuse)

Virtual machines capture the programming environment

- Package, *freeze*, and expose the environment
- VMWare, KVM, VirtualBox, Vagran,…

Lighter solutions (containers)

- Only capture software dependencies
- Docker, Rocket, OpenVZ, LXC, Conda

Capturing the command-line history, input/output, specification

CDE, ReproZip (NewYork University)

Outline

- Context
- Levels of reproducibility
- Reproducibility-friendly features
- Open Challenges

1. From repeat to replicate

Automatically finding the right set *of compatible libraries*

- Docker, VM allows to freeze the environment → Need to liquefy!
- Given a program P that can be repeated in an environment E…
- … Find an environment E’ (E’ uses more recent versions of libraries than E) where P still works

2. From repeat to reuse: Querying

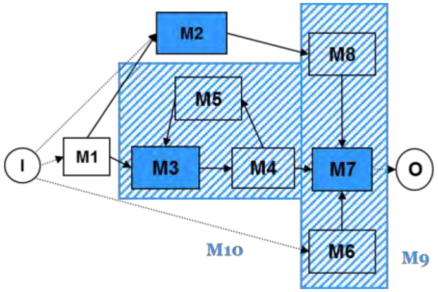
- Workflow Repositories queried (IR-style)
- Open question: Query languages for repositories
  - Given a input and/or and output format/type
  - *Given a workflow – find similar workflows*
  - …
- Core of the problem: Workflow similarity
  - State-of-the-art [SCB+14]
  - Need to design hybrid and efficient solutions

- Same point with Reproducible papers (Notebooks)
  - Interactive computational environment
  - Combination of code execution, text, mathematics, plots and rich media into a single document

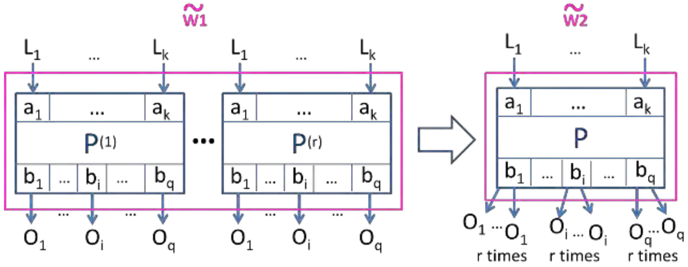
→Efficiently reusing (searching for) Notebooks is an open question

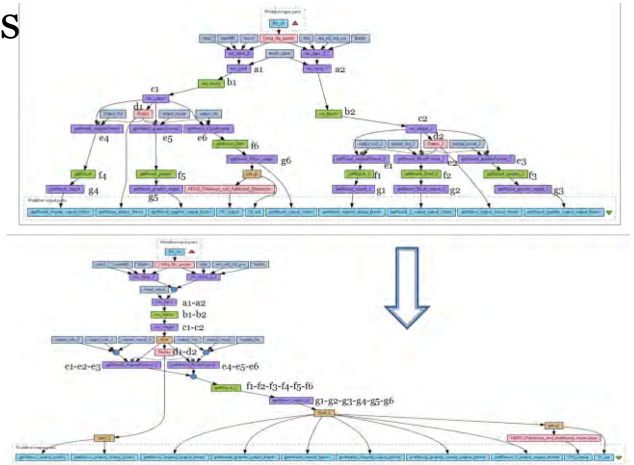
3. From repeat to reuse: Reduce the complexity of workflow structure

- Designing more coarse-grained workflows
  - **Biton *et al.* :** Automatic Design of subworkflows (graph-based)
  - **Alper *et al.***: **Abstraction** of provenance traces
  - **Gaignard *et al.*: Summarization** (Web Semantics)



- **Refactoring** workflows
  - Remove redundancies in workflows
    - **DistillFlow (Chen et al.)**: simplifying workflows : Rewritting **Anti-patterns,** Based on Taverna's semantics





Conclusion

- Too many scientific results are not reproducible
- Several Scientific workflow systems and companion tools are mature solutions
  - Repeat is (almost) always reachable
  - Next levels may be more difficult to reach
- Several open challenges are directly related to improvement in research in computer science (graphs, algorithmics…)
- Several Initiatives: Force 11, Data and Software Carpentry

software carpentry

elixir

FORCE11
The Future of Research Communications and e-Scholarship

DATA CARPENTRY
MAKING DATA SCIENCE MORE EFFICIENT

A series of ReproHackathons

* ReproHack1: RNA-Seq data from patients with uveal melanoma
* June 1-2, Gif s/Yvette, 25 participants (IGRoussy, Curie, Pasteur, Saclay, Paris, Nantes, Lyon, ...)

https://ifb-elixirfr.github.io/ReproHackathon/hackathon_1.html

Systems : SnakeMake, NextFlow, iPython notebooks, Galaxy, scripts...
Executed in the Cloud@IFB

Testing several levels of reproducibility: repeat and replicate

More soon!

ReproVirtuFlow @

Join us!

https://www.madics.fr/actions/actions-en-cours/reprovirtuflow/