



# Open Science and Artificial Intelligence in Research Software Engineering

**Lecturer: Daniel Garijo**

**ETSI Informáticos, Ontology Engineering Group,  
Universidad Politécnica de Madrid, Spain**

**<https://oeg.fi.upm.es/>**

**Session 5: Computational Methods\***

\*with slides from slides from **Yolanda Gil (Ed.) et. al.** The Scientific Paper of the Future

<https://scientificpaperofthefuture.org/> and

Data Science for Non-Programmers. Information Sciences Institute, University of Southern  
California. 2016. Credit: <http://www.datascience4all.org/>



daniel.garijo@upm.es



@dgarijov

This work is licensed under the license

**CC BY-NC-SA 4.0 International**

<http://purl.org/NET/rdflicense/cc-by-nc-sa4.0>



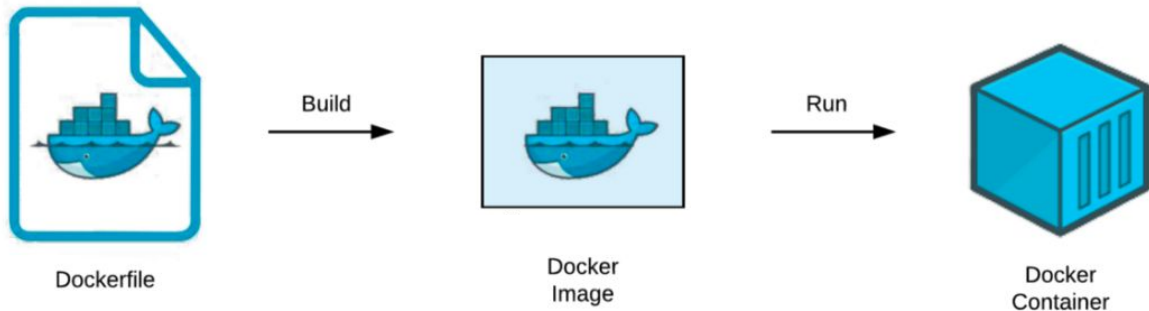
You are free:

- to Share — to copy, distribute and transmit the work
- to Remix — to adapt the work

Under the following conditions

- Non-commercial – You cannot use it for commercial purposes, nor for training inside a commercial company
- Attribution — You must attribute the work by inserting
- “[source <http://www.oeg-upm.net/>]” at the footer of each reused slide
- A credits slide stating: “These slides are partially based on ‘Open Science and Artificial Intelligence in Research Software Engineering: Introduction’ by Daniel Garijo and Oscar Corcho”
- Share-Alike

Previously on OS and AI in RSE...



**Instructions** for  
creating an image

File **ready for  
execution**

**Execution** of an  
image (you can run  
the same image  
multiple times)

What if you want to synchronize multiple services?

- Not a good practice to have all in the same container
- Multiple containers can talk to each other
  - Shared volumes (risky)
  - Ports/APIs (cleaner solution)
- Docker compose is an orchestrator



Executable File | 29 lines (29 sloc) | 624 Bytes

```
1  version: "2"
2  services:
3    #Morph-csv
4    morphcsv:
5      container_name: morphcsv
6      image: oegdataintegration/morph-csv:1.0.1
7      shm_size: '16gb'
8      volumes:
9        - ./data:/data
10       - ./results:/results
11       - ./mappings:/mappings
12       - ./queries:/queries
13       - ./tmp/csv:/morphcsv/tmp/csv
14      restart: always
15      depends_on:
16        - postgres
```

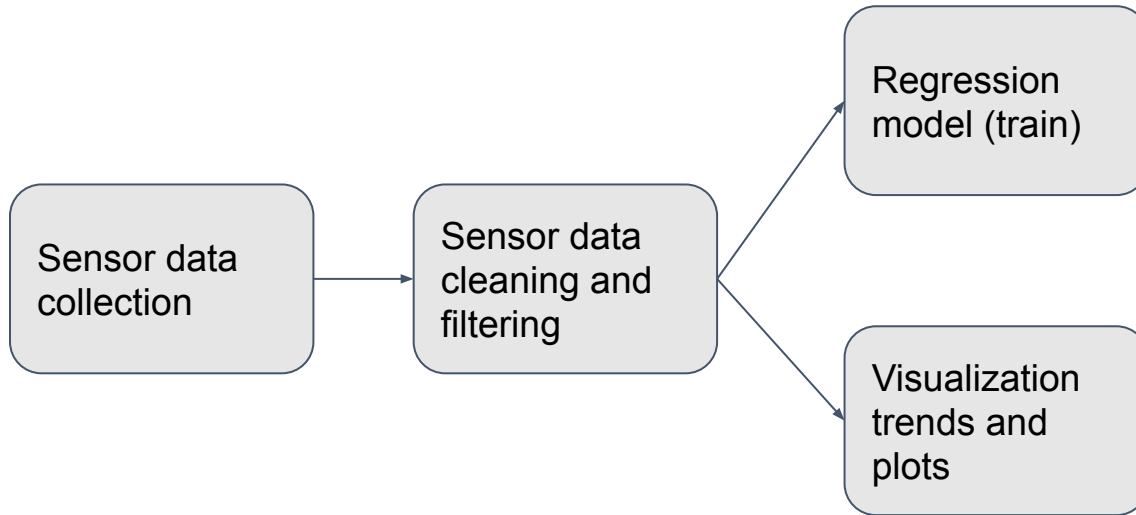
docker-compose.yml

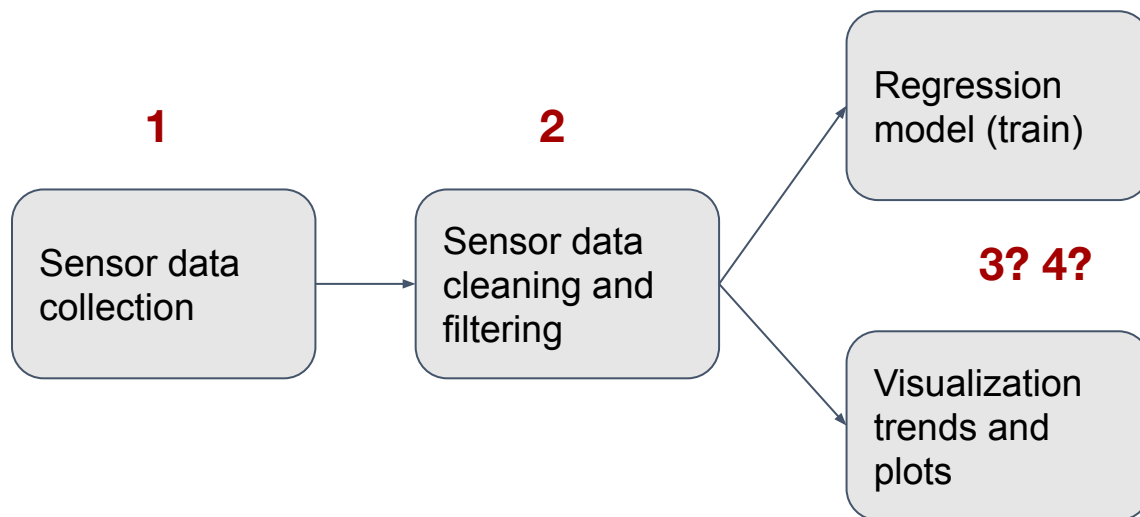
```
17  #Postgres
18  postgres:
19    container_name: postgres
20    image: postgres
21    restart: always
22    shm_size: '16gb'
23    environment:
24      POSTGRES_PASSWORD: csv
25      POSTGRES_USER: user
26      POSTGRES_DB: morphcsv
27    volumes:
28      - ./tmp/csv:/tmp/csv
29    restart: always
```

File orchestrating the full execution of all images:

- > `docker-compose up -d`: starts all services and sets up volumes
- > `docker-compose start/stop`: start/stop all services
- > `docker-compose rm`: removes stopped service containers (not volumes)

# What if I want to run multiple programs in order?





- What do the arrows mean?
- Is this figure clear?
  - Dataflow? Order?
- How would you synchronize components and dataflow?

3? 4?



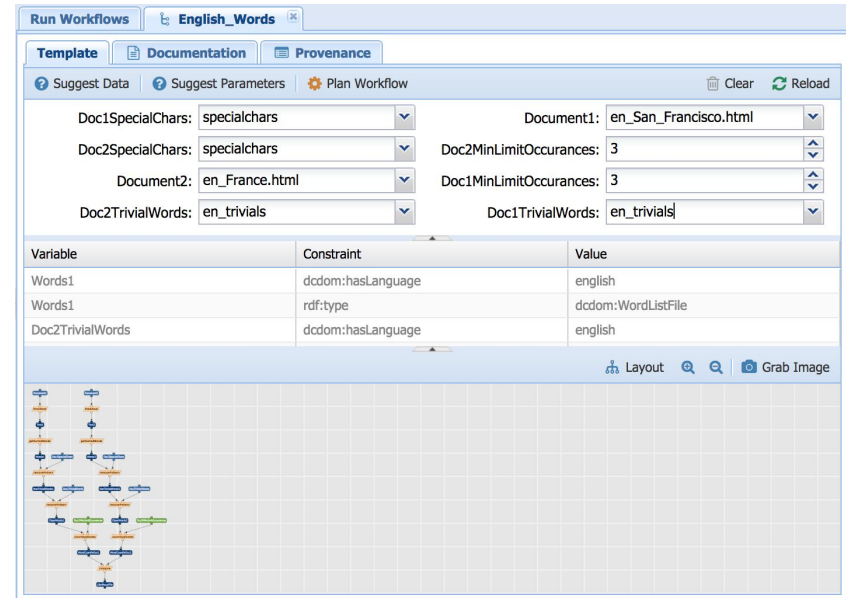
- Motivation
- Workflows
- Computational workflows in detail
- Workflow sketching
- Workflow systems
- The WINGS workflow system

Install WINGS (the image is a little heavy, 1.82 GB)

See <https://github.com/KnowledgeCaptureAndDiscovery/wings>

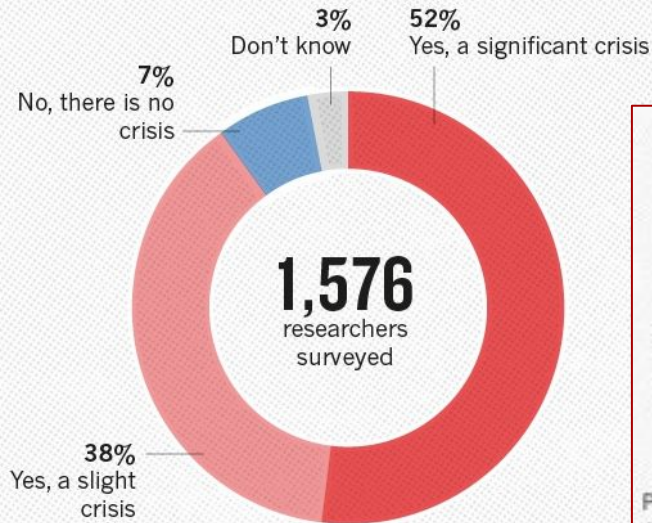
1. Download docker-compose.yml
2. Download portal.properties
3. run docker compose up

It will take a while (1.82 GB)



- **Motivation**
- Workflows
- Computational workflows in detail
- Workflow sketching
- Workflow systems
- The WINGS workflow system

## IS THERE A REPRODUCIBILITY CRISIS?



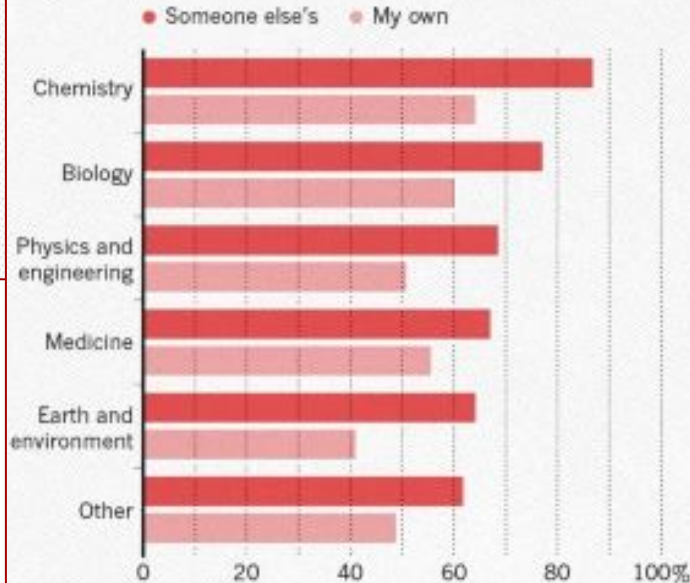
## Retraction Watch

Tracking retractions as a window into the scientific process



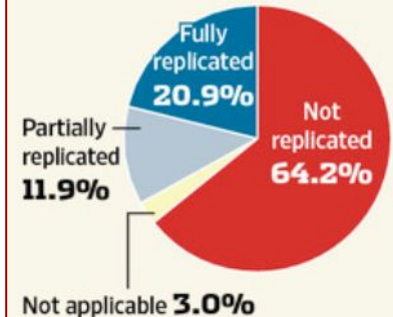
## HAVE YOU FAILED TO REPRODUCE AN EXPERIMENT?

Most scientists have experienced failure to reproduce results.



## No Cure

When Bayer tried to replicate results of 67 studies published in academic journals, nearly two-thirds failed.



Source: Nature Reviews Drug Discovery

<https://www.nature.com/articles/533452a>



- “**Ambiguity** in program descriptions leads to the possibility, if not the certainty, that a given natural language description can be converted into computer code in various ways, each of which may lead to different numerical outcomes.” [Ince et al 2012]
- Analysis of 18 quantitative papers published in Nature Genetics in the past two years found that reproducibility was not achievable even in principle in 10 cases, **even when datasets are published** [Ioannidis et al 2009]
- “Data processing, however, is often not described well enough to allow for exact reproduction of the results, leading to exercises in ‘**forensic bioinformatics**’ where aspects of raw data and reported results are used to infer what methods must have been employed.” [Baggerly and Coombes 2009]

[Ince et al 2012] Ince, D., Hatton, L. & Graham-Cumming, J. The case for open computer programs. *Nature* 482, 485–488 (2012). <https://doi.org/10.1038/nature10836>

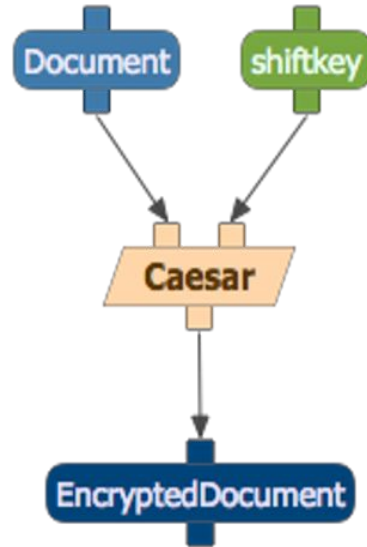
[Ioannidis et al 2009] Ioannidis JP, Allison DB, Ball CA, et al. Repeatability of published microarray gene expression analyses. *Nat Genet.* 2009; 41:149–155. doi: 10.1038/ng.295.

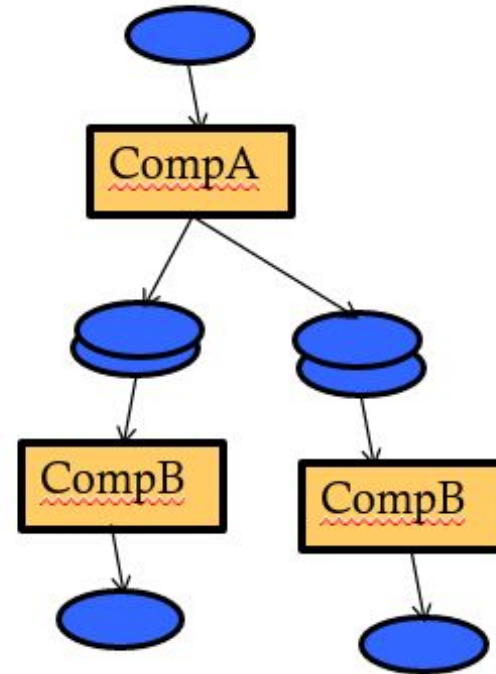
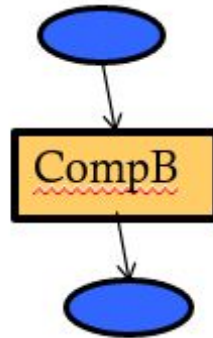
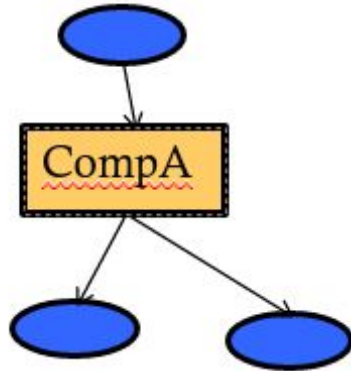
[Baggerly and Coombes 2009] Keith A. Baggerly, Kevin R. Coombes. "Deriving chemosensitivity from cell lines: Forensic bioinformatics and reproducible research in high-throughput biology." *Ann. Appl. Stat.* 3 (4) 1309 – 1334, December 2009. <https://doi.org/10.1214/09-AOAS291>

Credit: <http://www.datascience4all.org/>

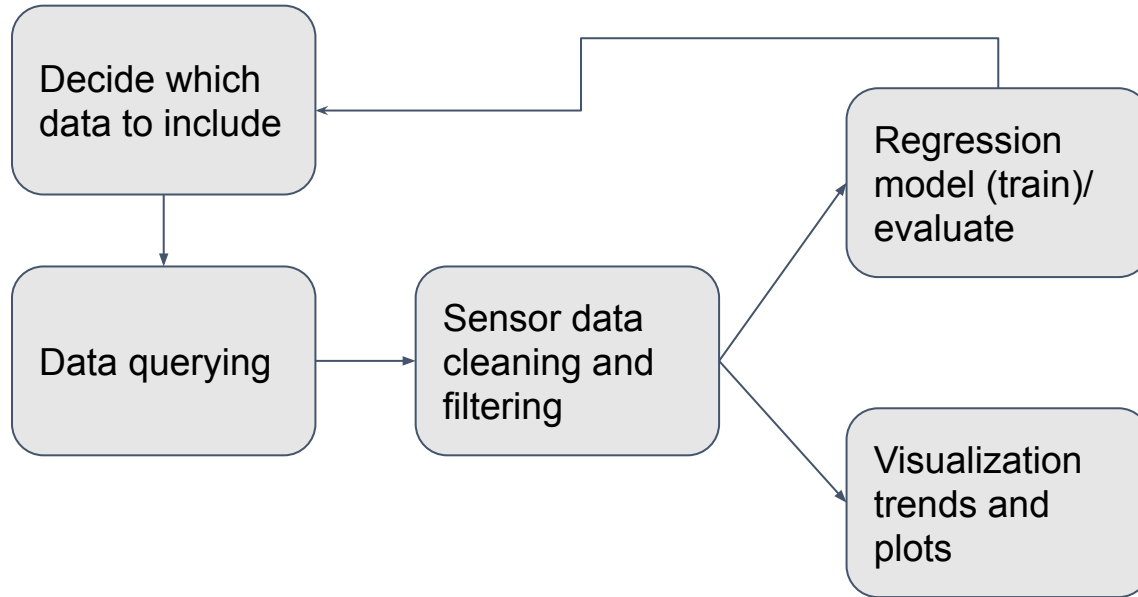
- Motivation
- **Workflows**
- Computational workflows in detail
- Workflow sketching
- Workflow systems
- The WINGS workflow system

## Inputs, outputs and parameters









## 1. Workflows of **human activities**

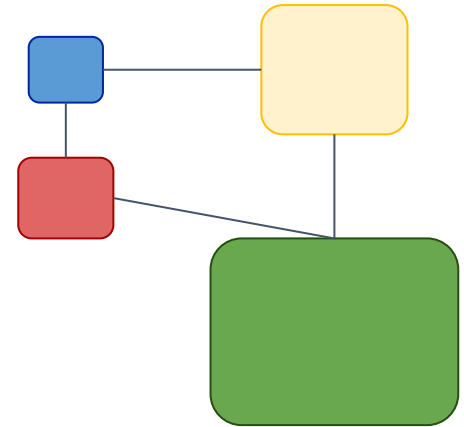
- E.g., checking patient in hospital
- E.g., sending a letter

## 2. Workflows of **web services**

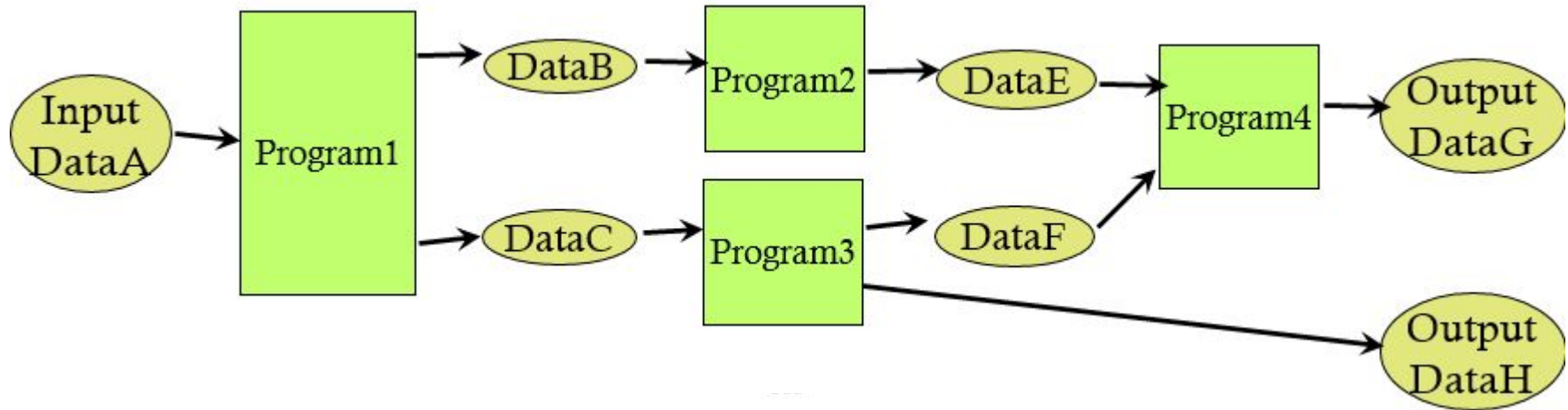
- E.g., integration of business services
- E.g., access Grobid for analyzing your pdf

## 3. Workflows **composed of programs**

- E.g., encrypt each sentence in a document
- E.g., clean and train your model



- Workflow is represented as a **graph of connected nodes**
  - Nodes represent programs and data (alternatively)
  - Links represent how data flows from program to program (output to input)
- Computational workflows are **compositions of programs**
  - No user interaction during execution
  - No cycles allowed!



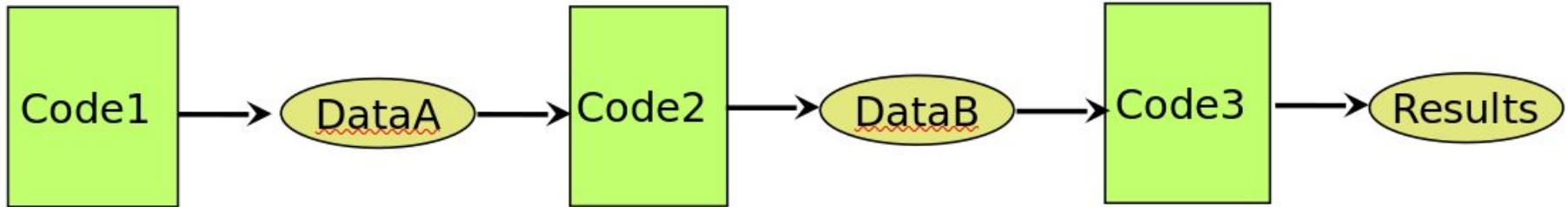
Credit: <http://www.datascience4all.org/>

Focus is on the work being done, not the results of the work

- Eg, you should start by checking in with the triage nurse, then talk to the admissions person about insurance, then wait in the lobby to be called, then a nurse will take your vitals, **then a doctor will see you**



Credit: <http://www.datascience4all.org/>



What types of workflows are the following?

- A GitHub action for generating documentation by a commit
- A machine learning pipeline
- A social experiment in a study performed in 200 users
- Cooking a recipe with a robot
- A laboratory protocol (in vitro)
- A laboratory protocol (in silico)
- Assembling furniture following instructions



What types of workflows are the following?

- A GitHub action for generating documentation by a commit
  - **Hybrid**
- A machine learning pipeline
  - **Computational**
- A social experiment in a study performed in 200 users
  - **Human**
- Cooking a recipe with a robot
  - **Hybrid**
- A laboratory protocol (in vitro)
  - **Hybrid** (may be computational)
- A laboratory protocol (in silico)
  - **Computational**
- Assembling furniture following instructions
  - **Human**



- **Data view:**

- E.g., “We take File1 and InitCond1 parameters, generate Prediction1, and use that to generate Visualization1”

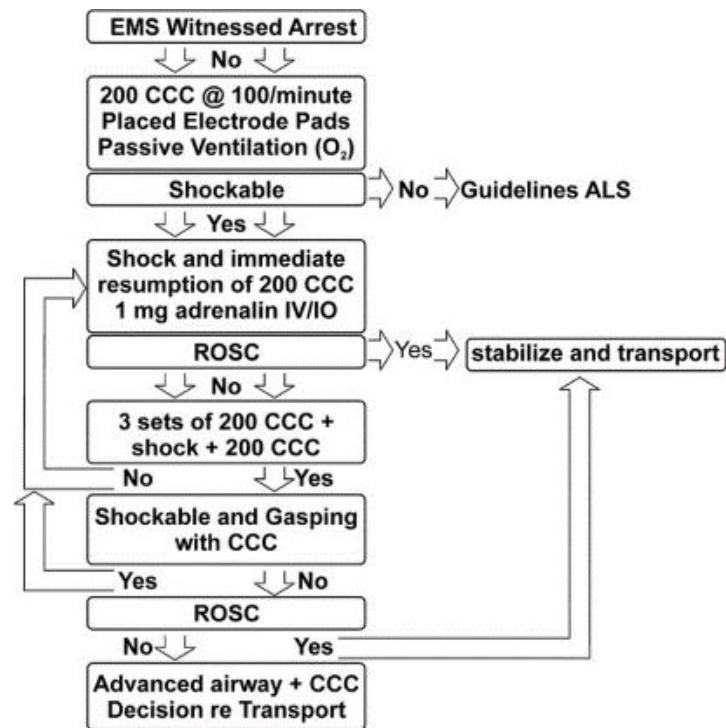
- **Step view:**

- E.g., “We start by removing word endings, then we calculate inverse word/document frequency, then we train a classifier, then we predict sentiment”

- **Execution view:**

- E.g., “We run the workflow under a range of assumptions about fertilizer policies, market conditions, and weather forecasts.”

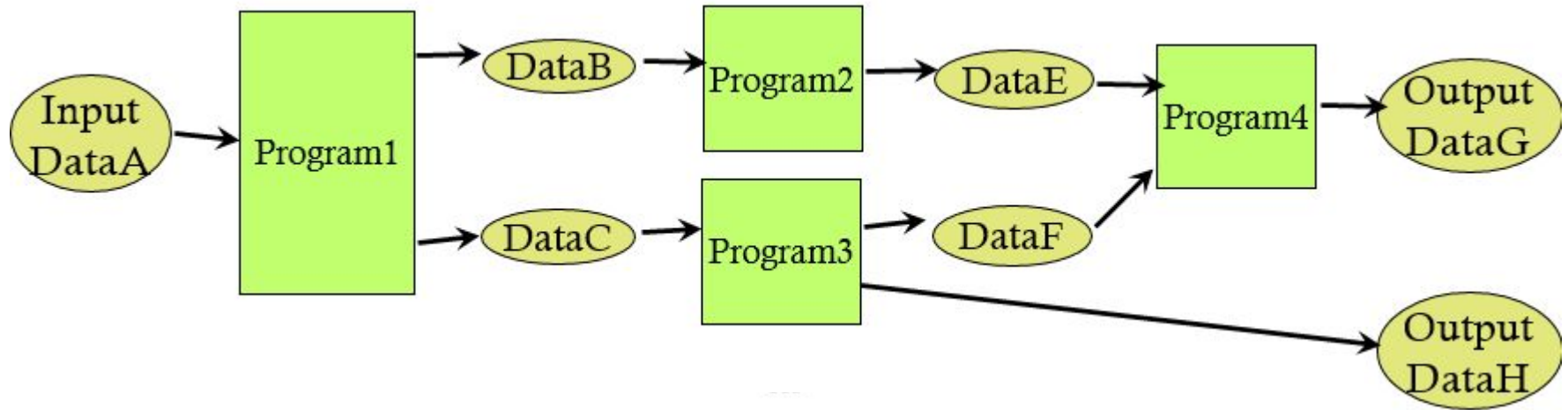
## Decision view:



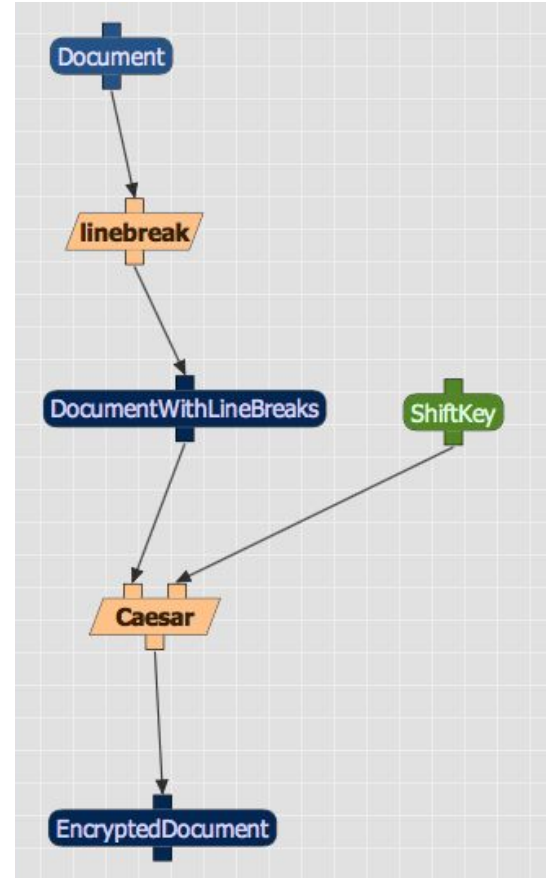
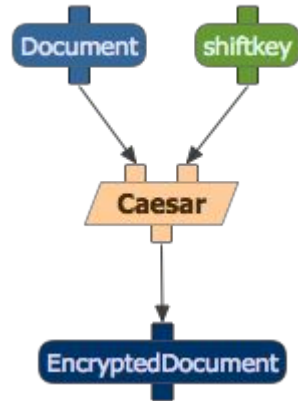


- Motivation
- Workflows
- **Computational workflows in detail**
- Workflow sketching
- Workflow systems
- The WINGS workflow system

# Computational Workflows in detail

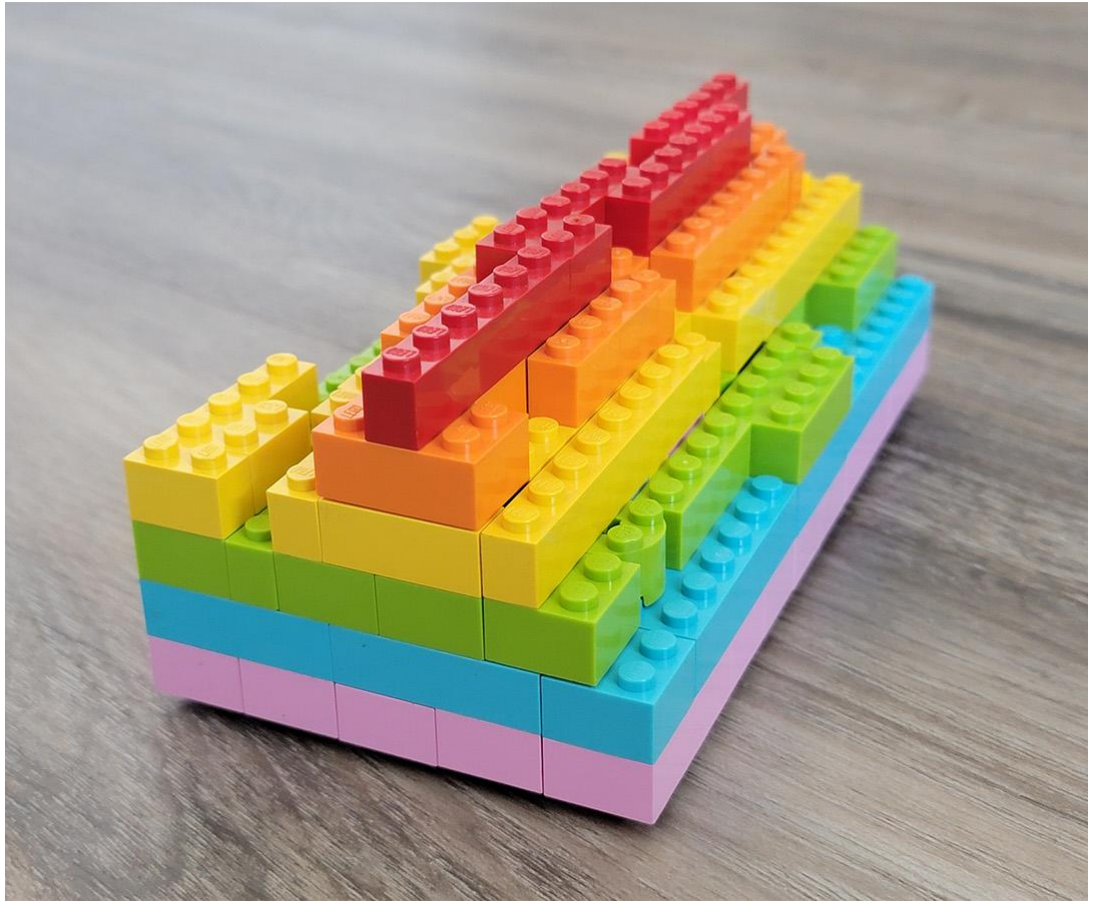


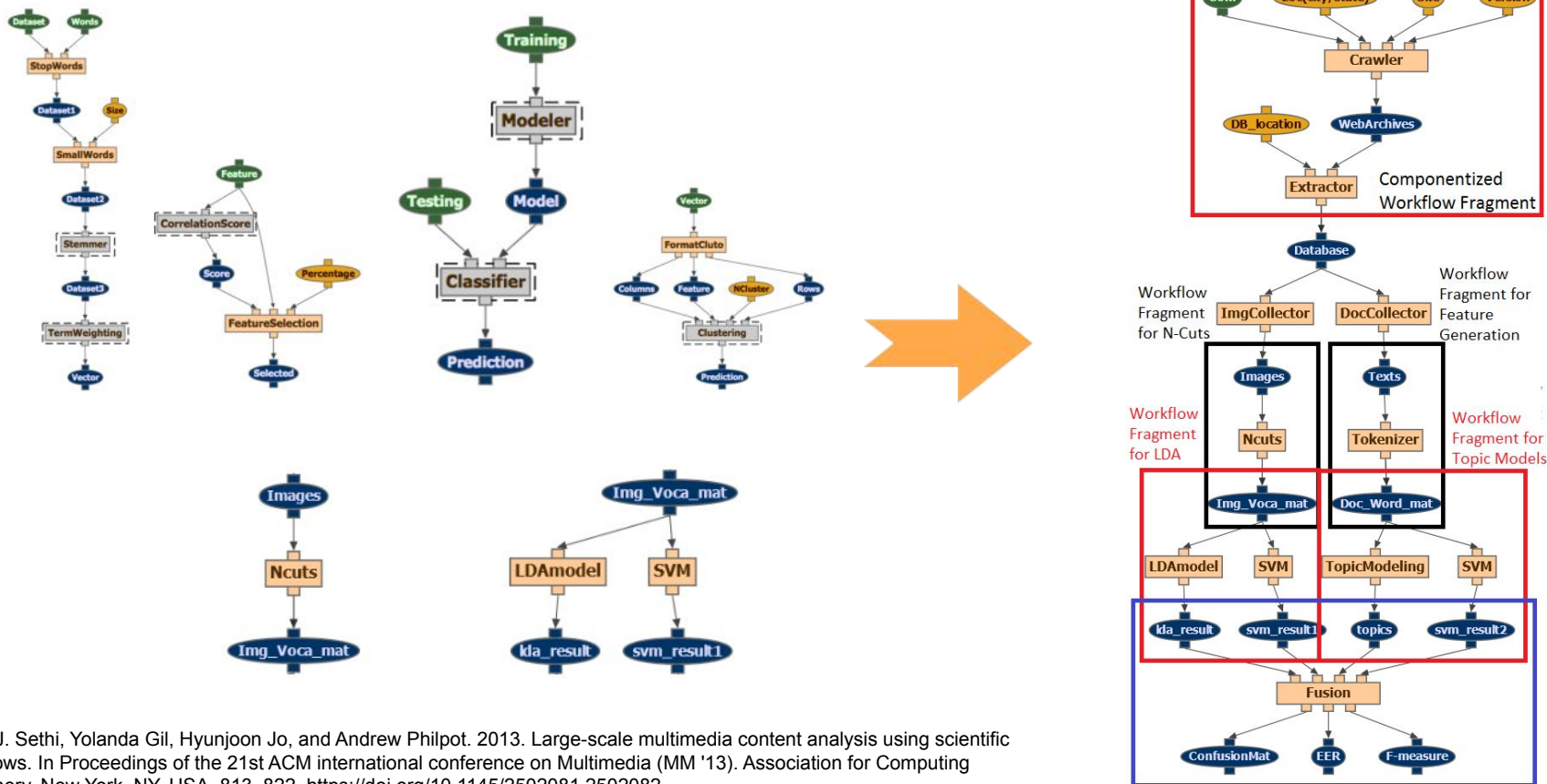
Credit: <http://www.datascience4all.org/>



Credit: <http://www.datascience4all.org/>

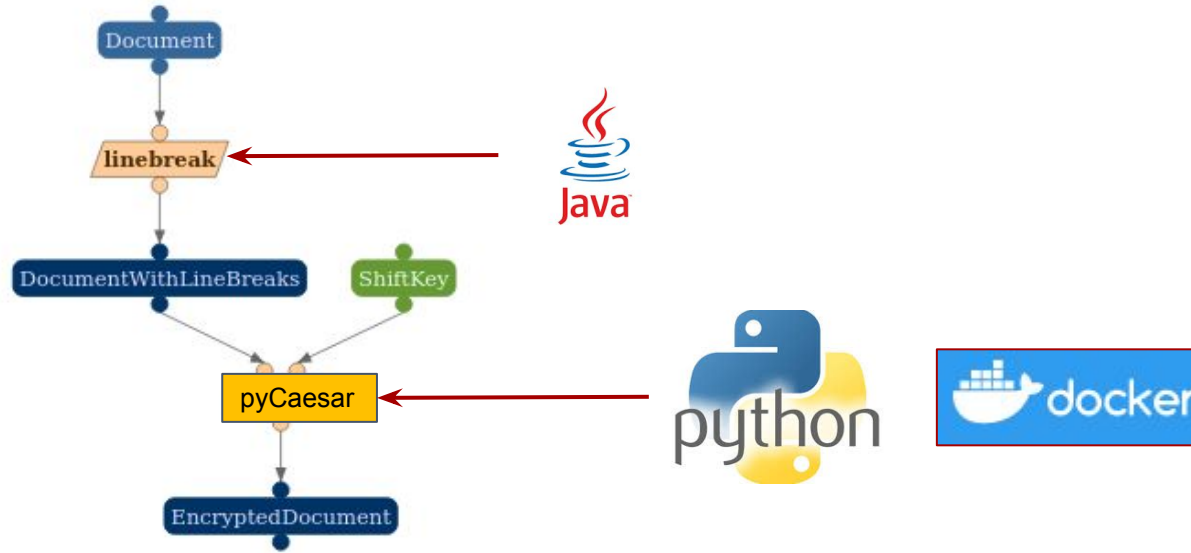
**Building blocks!**



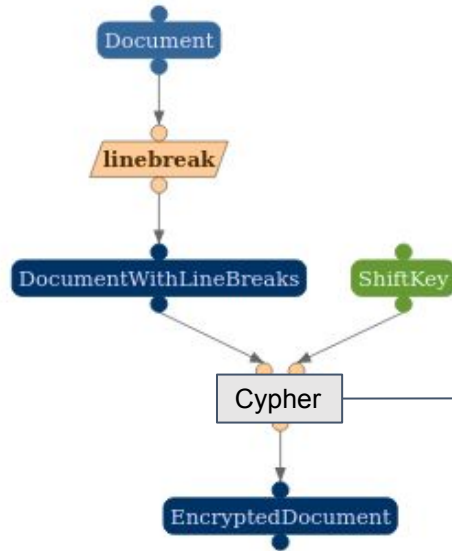


Ricky J. Sethi, Yolanda Gil, Hyunjoon Jo, and Andrew Philpot. 2013. Large-scale multimedia content analysis using scientific workflows. In Proceedings of the 21st ACM international conference on Multimedia (MM '13). Association for Computing Machinery, New York, NY, USA, 813–822. <https://doi.org/10.1145/2502081.2502082>

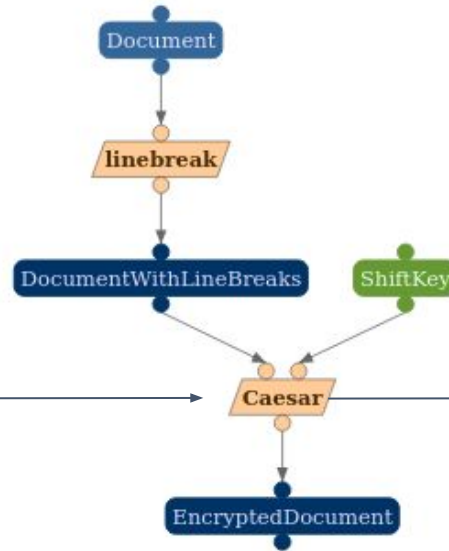
Credit: <http://www.datascience4all.org/>



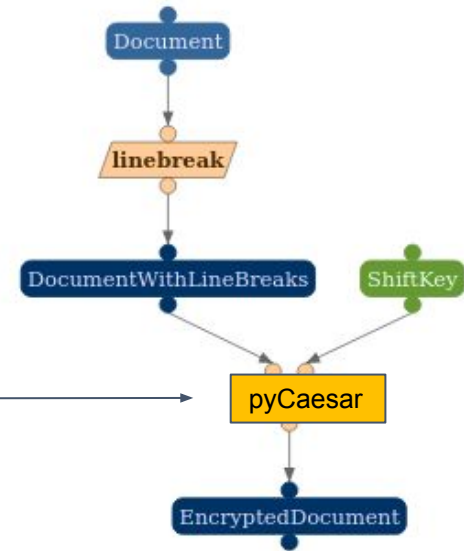
## Methods



## Algorithms



## Implementation





- Simple programming paradigm
- Modular assembly
- Composing heterogeneous code
- Abstraction
- Data preparation steps
- Data visualization steps
- Documenting provenance: reproducibility
- Automatic processing of multiple inputs
- Large-scale processing
- Facilitating communication across data science expertise areas

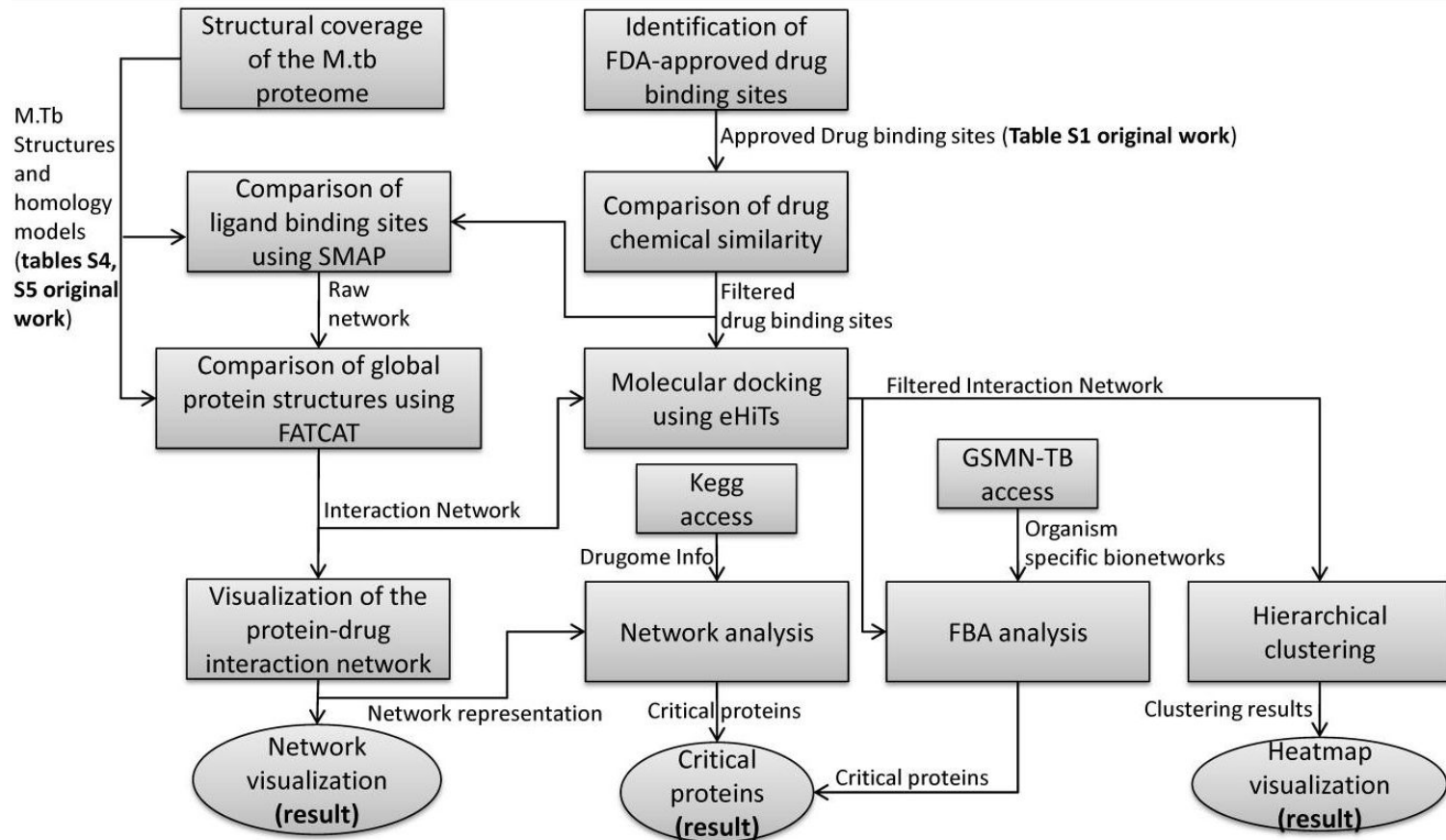


Credit: <http://www.datascience4all.org/>



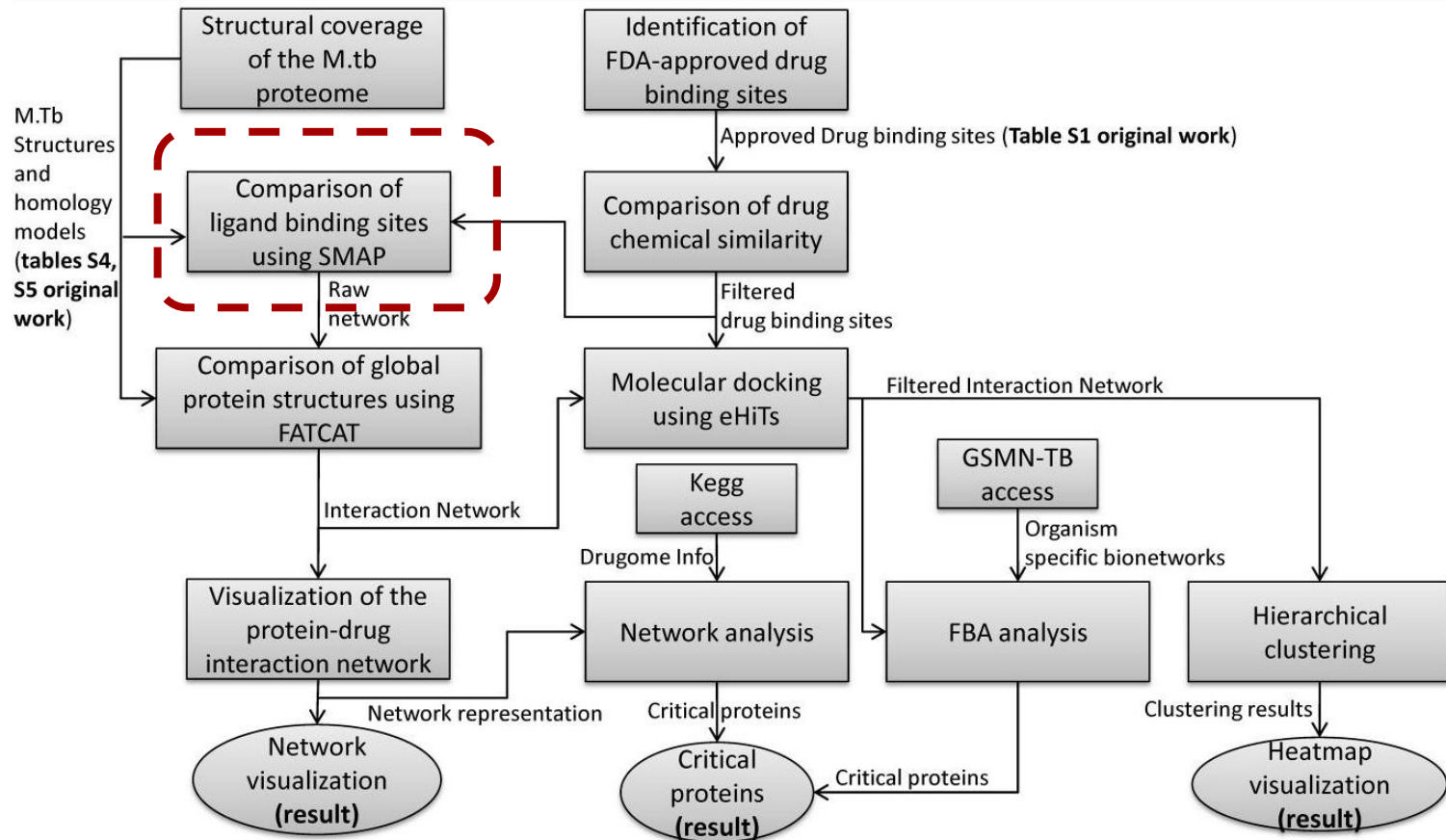
- Motivation
- Workflows
- Computational workflows in detail
- **Workflow sketching**
- Workflow systems
- The WINGS workflow system

# What the paper says versus what the experiment does (1)



Garijo D, Kinnings S, Xie L, Xie L, Zhang Y, Bourne PE, et al. (2013) Quantifying Reproducibility in Computational Biology: The Case of the Tuberculosis Drugome. PLoS ONE 8(11): e80278. <https://doi.org/10.1371/journal.pone.0080278>

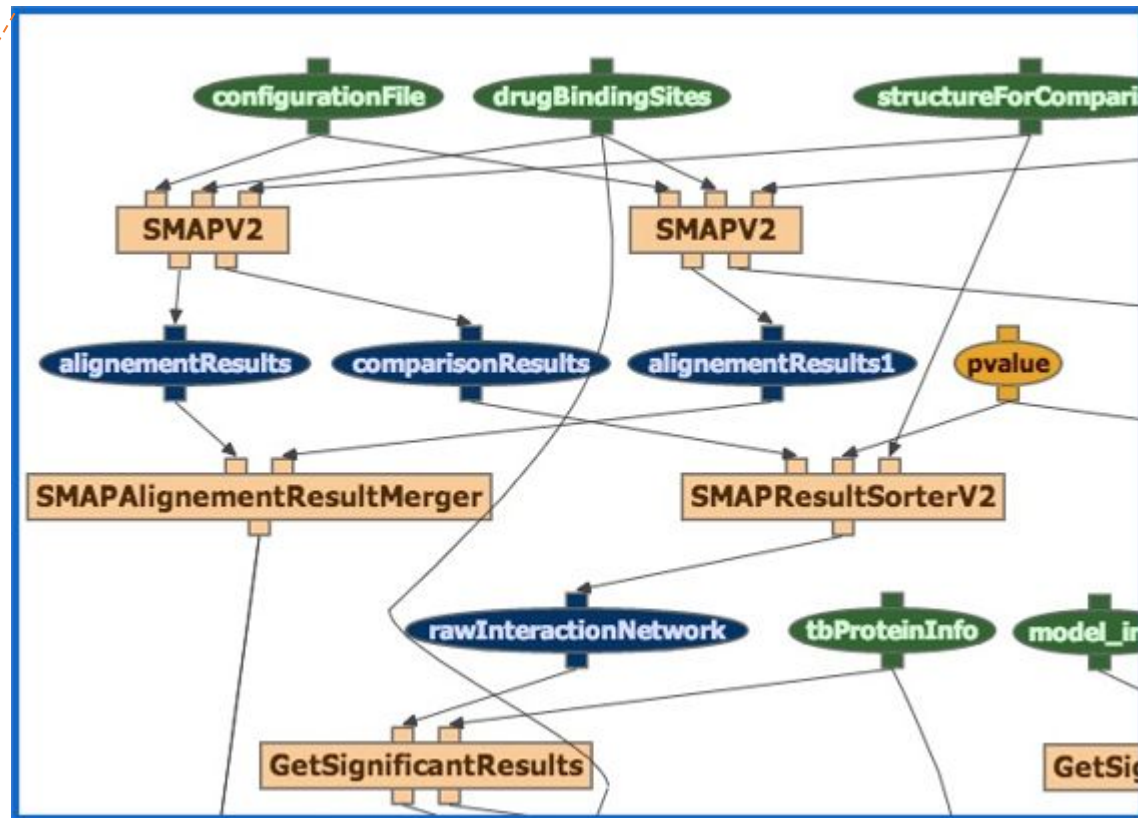
# What the paper says versus what the experiment does (1)



Garijo D, Kinnings S, Xie L, Xie L, Zhang Y, Bourne PE, et al. (2013) Quantifying Reproducibility in Computational Biology: The Case of the Tuberculosis Drugome. PLoS ONE 8(11): e80278. <https://doi.org/10.1371/journal.pone.0080278>

## What the paper says versus what the experiment does (2)

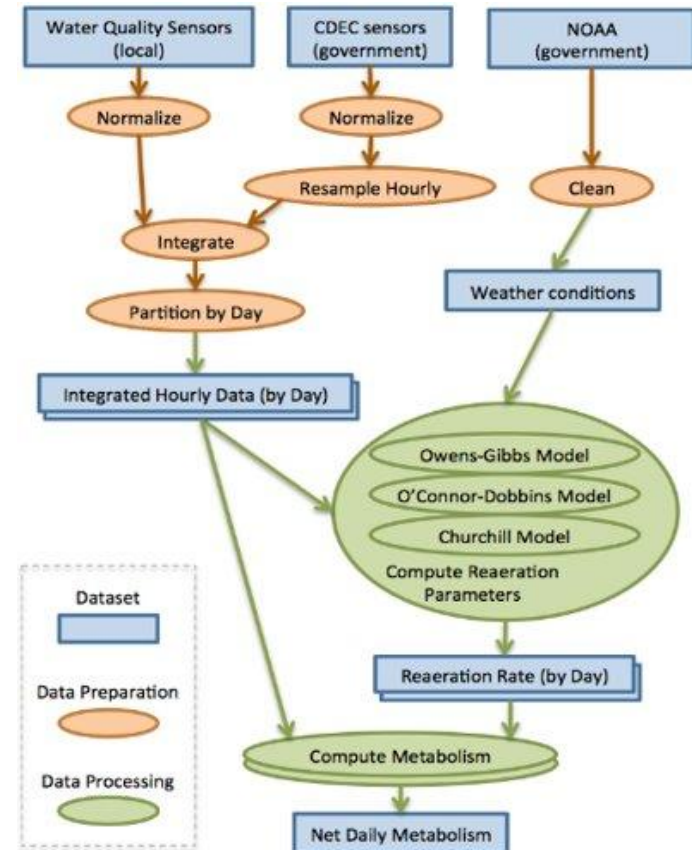
Comparison of ligand  
binding sites using  
SMAP



Garijo D, Kinnings S, Xie L, Xie L, Zhang Y, Bourne PE, et al. (2013) Quantifying Reproducibility in Computational Biology: The Case of the Tuberculosis Drugome. PLoS ONE 8(11): e80278. <https://doi.org/10.1371/journal.pone.0080278>

Credit: <http://www.datascience4all.org/>

1. Compile the **command line invocation** to all your codes
  - o Input data, parameters, configuration files
  - o Include data preparation codes
2. Consider how the **data flows** from code to code
3. Starting with the input data, work your way to **the results**
4. If any steps were done with **manual intervention**, indicate it in your diagram
5. Create **subworkflows** if it gets large
6. Add a **legend** to your figure
  - o Arrows and boxes should have a **clear meaning**



Credit: <http://www.datascience4all.org/>

Draw a step-by-step process of the workflow from the following text:

**[...] We took a quartzite sample from the Hellerman thrust zone, and cut 3 thin sections. We measured c-axis orientations using a petrographic microscope. We rotated to a common reference frame using Duyster's StereoNett program. We plotted the data on lower hemisphere, equal area projections using Duyster's StereoNett program, shown in Figure 4. [...]**

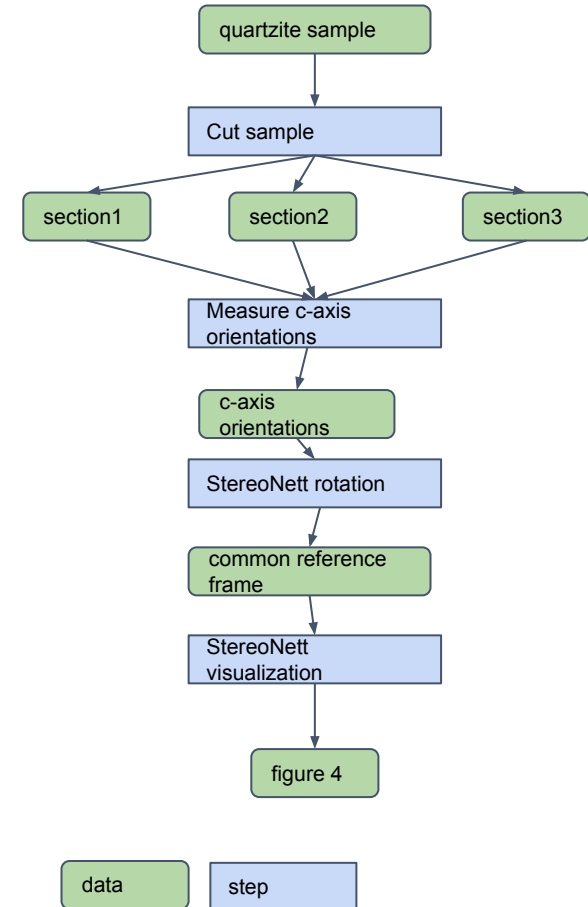


**Understanding kinematic data from the Hellerman thrust zone, Jade Silverstein**

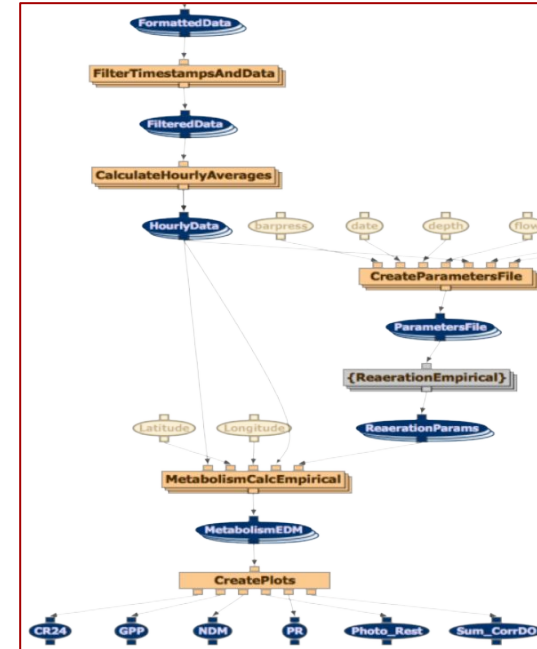
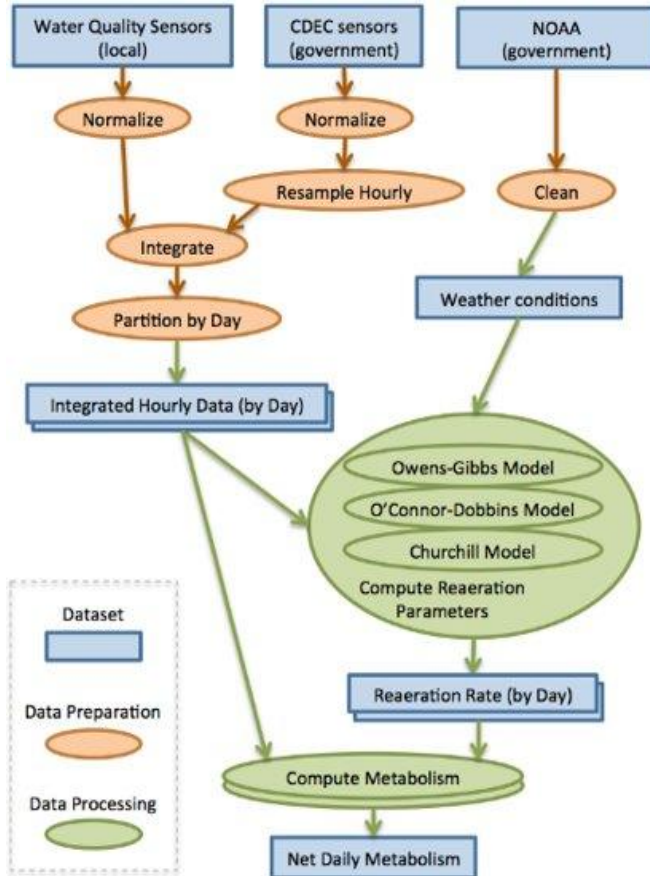
Draw a step-by-step process of the workflow you understand from the following text:

**[...] We took a quartzite sample from the Hellerman thrust zone, and cut 3 thin sections. We measured c-axis orientations using a petrographic microscope. We rotated to a common reference frame using Duyster's StereoNett program. We plotted the data on lower hemisphere, equal area projections using Duyster's StereoNett program, shown in Figure 4. [...]**

**Understanding kinematic data from the Hellerman thrust zone, Jade Silverstein**



# From a sketch to a computational workflow



Credit: <http://www.datascience4all.org/>



Draw a step-by-step process of the workflow in your first assignment

Some volunteers to discuss the results!



- Motivation
- Workflows
- Computational workflows in detail
- Workflow sketching
- **Workflow systems**
- The WINGS workflow system

- Many choices
  - Academic prototypes
  - Operational open source
  - Commercial
- Each has different capabilities
  - Workflow validation
  - Scalable computations
  - Domain functions



Credit: <http://www.datascience4all.org/>

- Record data, software, results, notes, etc.
  - Records what code was run when generating a result
  - Can re-run code with new data



Jupyter wikidata (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted SPARQL

### Which algorithms or formulas in Wikidata do not have an image yet?

```
In [1]: %endpoint http://query.wikidata.org/sparql
%display table
%show all

SELECT DISTINCT ?item ?itemLabel ?formula WHERE {
  {
    SELECT DISTINCT ?item ?formula WHERE {
      { ?item ((wdt:P31*/wdt:P279) wd:Q8366. ) UNION { ?item wdt:P2534 ?formula. }
      FILTER (NOT EXISTS { ?item wdt:P18 ?image. })
      FILTER (NOT EXISTS { ?item wdt:P31 wd:Q1266546. })
      FILTER (NOT EXISTS { ?item wdt:P373 ?category. })
    }
    LIMIT 5
  }
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }
}
ORDER BY ASC(?item)
```

Endpoint set to: <http://query.wikidata.org/sparql>  
Display: table  
Result maximum size: unlimited

item	itemLabel	formula
<a href="http://www.wikidata.org/entity/Q116076">http://www.wikidata.org/entity/Q116076</a>	CORDIC	
<a href="http://www.wikidata.org/entity/Q130762">http://www.wikidata.org/entity/Q130762</a>	multiplication algorithm	
<a href="http://www.wikidata.org/entity/Q140770">http://www.wikidata.org/entity/Q140770</a>	General number field sieve	
<a href="http://www.wikidata.org/entity/Q71746">http://www.wikidata.org/entity/Q71746</a>	Trachtenberg system	
<a href="http://www.wikidata.org/entity/Q93593">http://www.wikidata.org/entity/Q93593</a>	common subexpression elimination	

Total: 5, Shown: 5

Feature	Workflow systems	Electronic notebooks
Simple programming paradigm	✓	✓
Modular assembly	✓	✗
Composing heterogeneous codes	✓	✗ (limited)
Abstraction	✓	✗
Data preparation steps	✓	✓
Data visualization steps	✓	✓
Documenting provenance	✓	✓ (limited)
Automatic processing of multiple inputs	✓	✗
Large scale processing	✓	✗ (through technologies like Spark)
Facilitating communication across data science expertise areas	✓	✗

Credit: <http://www.datascience4all.org/>

- Motivation
- Workflows
- Computational workflows in detail
- Workflow sketching
- Workflow systems
- **The WINGS workflow system**

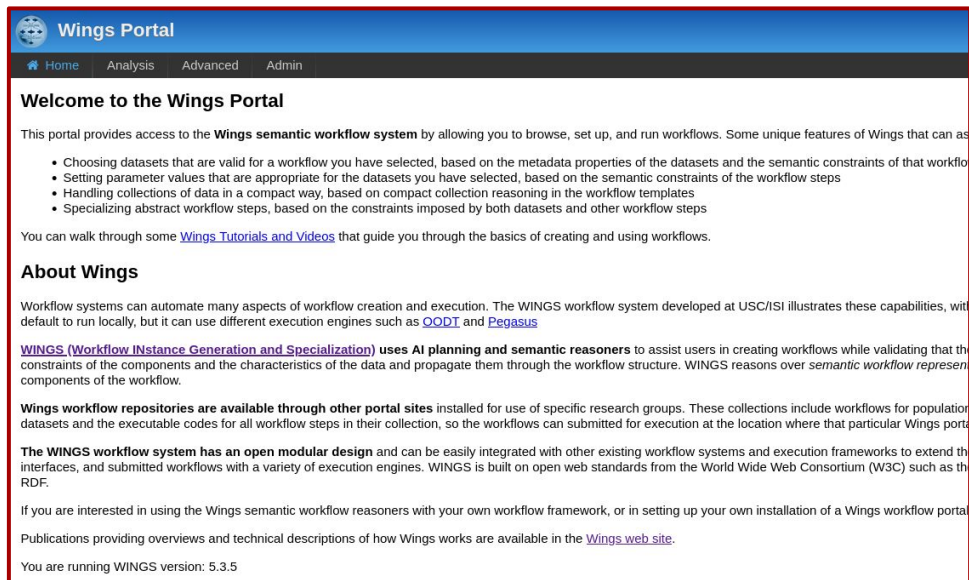
<https://www.wings-workflows.org/tutorial/tutorial.html>

git clone <https://github.com/KnowledgeCaptureAndDiscovery/wings.git>

```
docker-compose up -d
```

<http://localhost:8080/wings-portal>

user/pass: admin/admin123



The screenshot shows the WINGS Portal web interface. At the top is a blue header with the 'Wings Portal' logo and a navigation bar with links for Home, Analysis, Advanced, and Admin. Below the header, a 'Welcome to the Wings Portal' section contains a paragraph about the portal's purpose and a bulleted list of features: choosing valid datasets, setting parameter values, handling data collections, and specializing workflow steps. This is followed by a link to 'Wings Tutorials and Videos'. The 'About Wings' section describes the workflow system's capabilities, its development at USC/ISI, and its use of AI planning and semantic reasoners. It also mentions that workflow repositories are available through other portal sites and that the system has an open modular design. The interface concludes with a link to the 'Wings web site' for publications and a version notice: 'You are running WINGS version: 5.3.5'.

**Wings Portal**

Home Analysis Advanced Admin

### Welcome to the Wings Portal

This portal provides access to the **Wings semantic workflow system** by allowing you to browse, set up, and run workflows. Some unique features of Wings that can as

- Choosing datasets that are valid for a workflow you have selected, based on the metadata properties of the datasets and the semantic constraints of that workflow
- Setting parameter values that are appropriate for the datasets you have selected, based on the semantic constraints of the workflow steps
- Handling collections of data in a compact way, based on compact collection reasoning in the workflow templates
- Specializing abstract workflow steps, based on the constraints imposed by both datasets and other workflow steps

You can walk through some [Wings Tutorials and Videos](#) that guide you through the basics of creating and using workflows.

### About Wings

Workflow systems can automate many aspects of workflow creation and execution. The WINGS workflow system developed at USC/ISI illustrates these capabilities, with default to run locally, but it can use different execution engines such as [OODT](#) and [Pegasus](#).

[WINGS \(Workflow Instance Generation and Specialization\)](#) uses **AI planning and semantic reasoners** to assist users in creating workflows while validating that the constraints of the components and the characteristics of the data and propagate them through the workflow structure. WINGS reasons over *semantic workflow representations* of the workflow.

**Wings workflow repositories are available through other portal sites** installed for use of specific research groups. These collections include workflows for population datasets and the executable codes for all workflow steps in their collection, so the workflows can be submitted for execution at the location where that particular Wings portal is installed.

**The WINGS workflow system has an open modular design** and can be easily integrated with other existing workflow systems and execution frameworks to extend their interfaces, and submitted workflows with a variety of execution engines. WINGS is built on open web standards from the World Wide Web Consortium (W3C) such as the RDF.

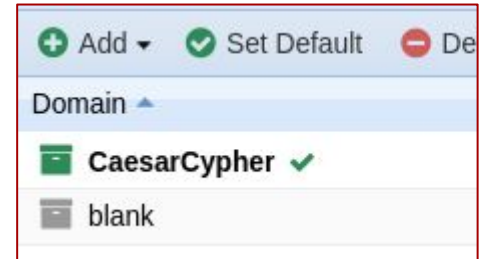
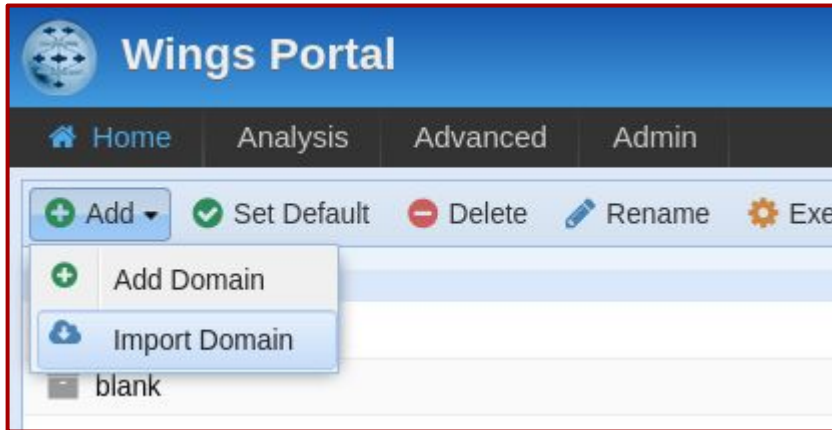
If you are interested in using the Wings semantic workflow reasoners with your own workflow framework, or in setting up your own installation of a Wings workflow portal, publications providing overviews and technical descriptions of how Wings works are available in the [Wings web site](#).

You are running WINGS version: 5.3.5

Start by importing a domain:

<https://www.wings-workflows.org/domains/students/CaesarCypher.zip>

Advanced -> Manage domain -> Import domain





Let's design a computational workflow with WINGS

- WINGS components
  - Abstract
  - Concrete
    - Components may even have Docker images!
- WINGS data
  - Hierarchy
  - Metadata
- Creating workflows
  - Abstract
  - Concrete
- Browsing results
- Creating rules (example for data selection, e.g., language selection)

## Run a workflow (Caesar's cypher)

- Run a workflow with your own dataset
- Edit a component (create a copy) and add a new parameter with a shift-key multiplier
- Create a component (take 2 files, count the total words, create a file with the result)
- Create a workflow with your component
- Add a rule? (if time)



Follow the **best practices** taught in class to perform an analysis over **10 open-access** articles using Grobid (or other text analysis tools). Your program should:

1. Draw a **keyword cloud** based on the abstract information
2. Create a visualization showing the **number of figures per article**.
3. Create a **list of the links** found in each paper.

You should explain (in your repository documentation) how you have validated each of your answers. **Create a document called "rationale.md" for this purpose**

### Steps:

- 1) Make pipeline with Grobid and initial selection of papers
- 2) Create Python scripts for addressing the questions
  - **Deadline: Feb, 15th**
- 3) Create documentation and an environment for running your experiments
  - **Deadline: Feb, 22nd**
- 4) Dockerize experiment.
  - **Deadline March, 1st**

Deadline for full individual practice: March 8th, **2023 (11:59 pm)**

## Checklist for individual project:

- Answer all questions (1..3)
- Have **reproducible** instructions on **how to set up an environment**
  - Do not upload the environment to GitHub
  - It should work with 10 pdfs different from yours
- Proper **documentation** (readthedocs + readme)
  - Readme should not be empty
- Zenodo integration (with a badge in the readme)
  - With a release in GitHub
- Proper metadata (codemeta file)
  - CFF file
- **Tests + continuous integration**
- **Dockerfile + docker run instructions**



## Install Spark

([https://spark.apache.org/docs/latest/api/python/getting\\_started/install.html](https://spark.apache.org/docs/latest/api/python/getting_started/install.html) )

Parallel processing with workflows!





# Open Science and Artificial Intelligence in Research Software Engineering

**Lecturer: Daniel Garijo**

**ETSI Informáticos, Ontology Engineering Group,  
Universidad Politécnica de Madrid, Spain**

**<https://oeg.fi.upm.es/>**

**Session 4:** Preserving computational environments using  
software containers\*

\*with slides from slides from David Chaves, Carlos Badenes  
and Esteban Gonzalez



**daniel.garijo@upm.es**



**@dgarijov**