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

License

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:

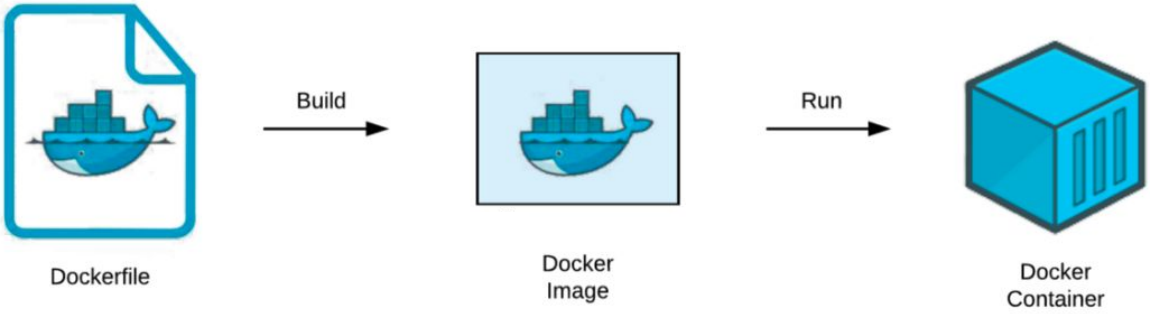
- to Share — to copy, distribute and transmit the work
- to Remix — to adapt the work

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…

Steps for creating a container



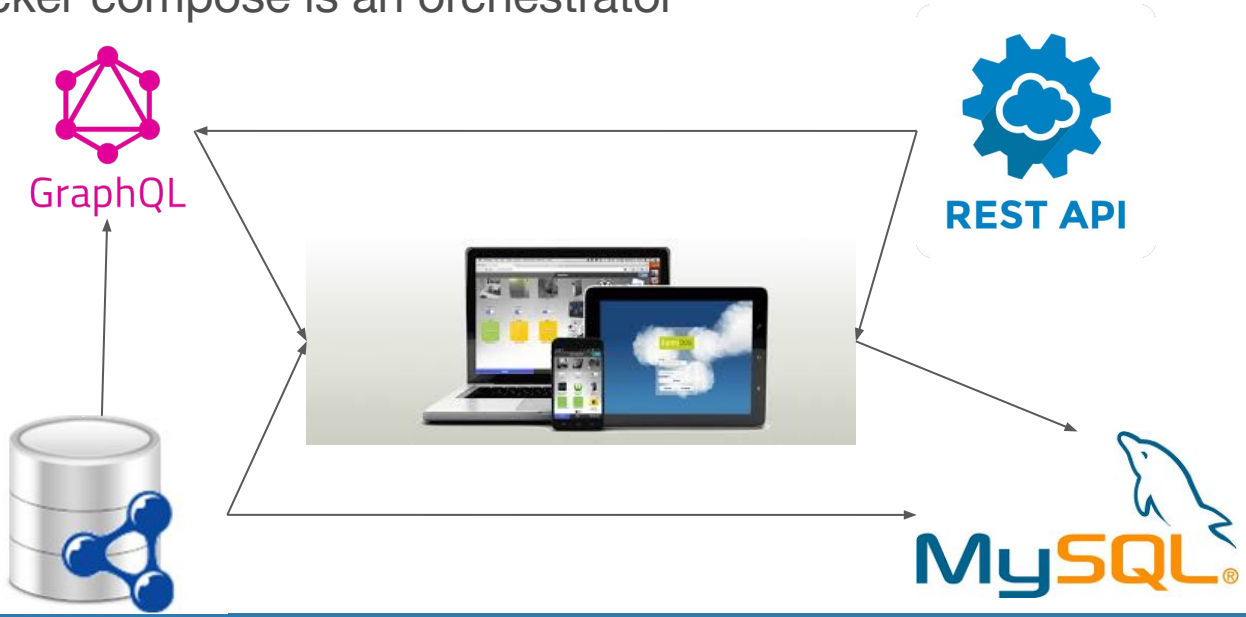
Instructions for creating an image

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



Docker compose: An example

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

docker-compose.yml

```yaml
version: "2"
services:
#Morph-csv
  morphcsv:
    container_name: morphcsv
    image: oegdataintegration/morph-csv:1.0.1
    shm_size: '16gb'
    volumes:
      - ./data:/data
      - ./results/:/results
      - ./mappings:/mappings
      - ./queries:/queries
      - ./tmp/csv:/morphcsv/tmp/csv
    restart: always
    depends_on:
      - postgres
```

```yaml
#Postgres
  postgres:
    container_name: postgres
    image: postgres
    restart: always
    shm_size: '16gb'
    environment:
      POSTGRES_PASSWORD: csv
      POSTGRES_USER: user
      POSTGRES_DB: morphcsv
    volumes:
      - ./tmp/csv:/tmp/csv
    restart: always
```

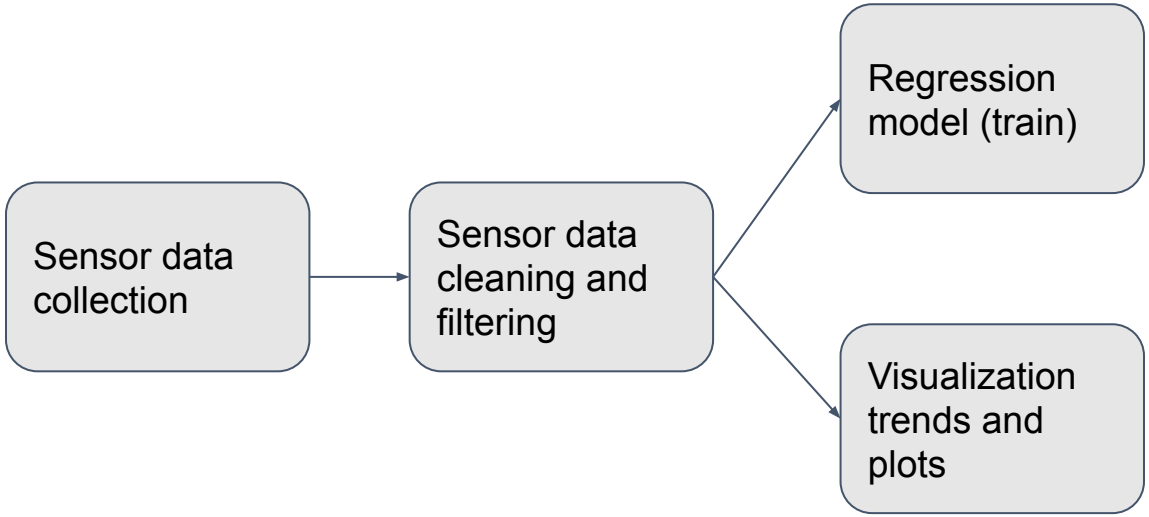
File orchestrating the full execution of all images:

> `docker-compose up -d`: starts all services and sets up volumes

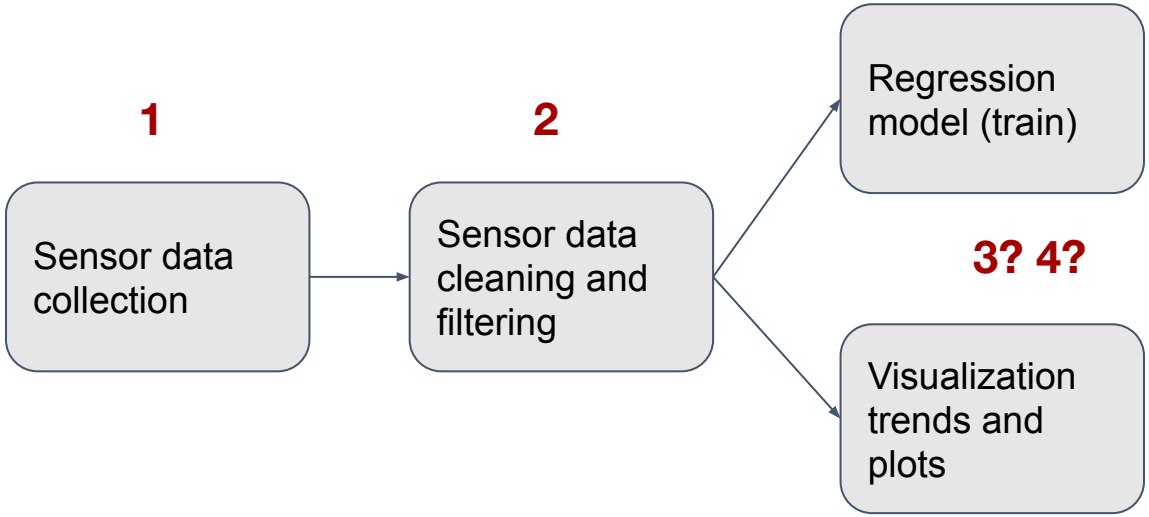
> `docker-compose start/stop`: start/stop all services

> `docker-compose rm`: removes stopped service containers (not volumes)

What if I want to run multiple programs in order?



What if I want to run multiple steps in order?



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

Outline for today

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

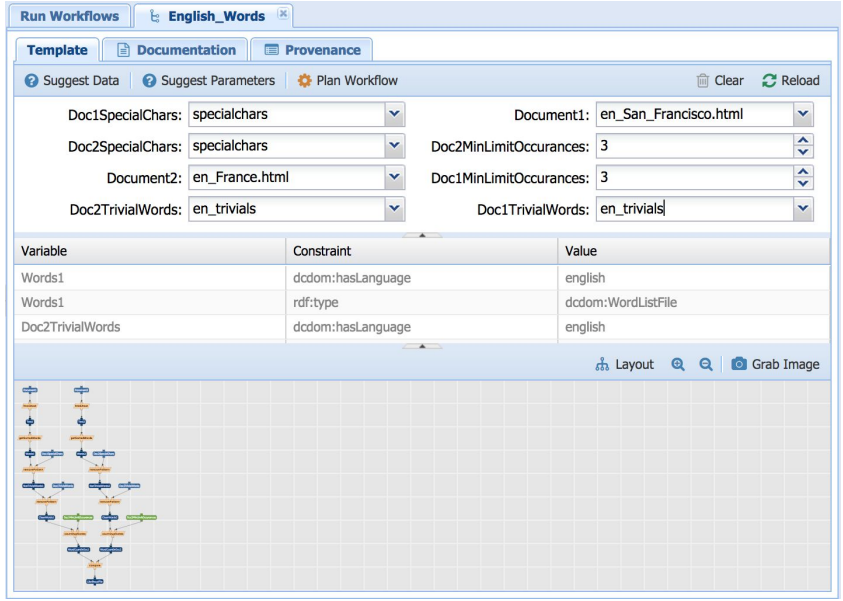
Before we get started!

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

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

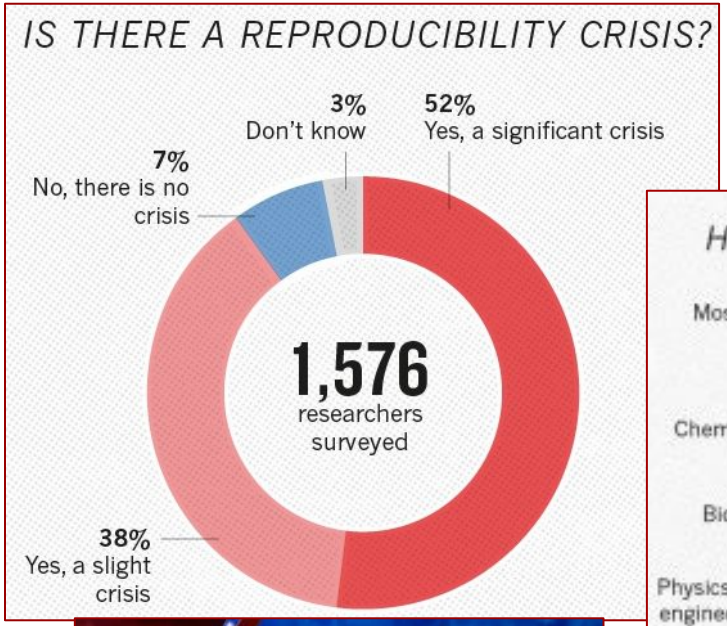
1. Download docker-compose.yml
2. Download portal.properties
3. run docker compose up

It will take a while (1.82 GB)



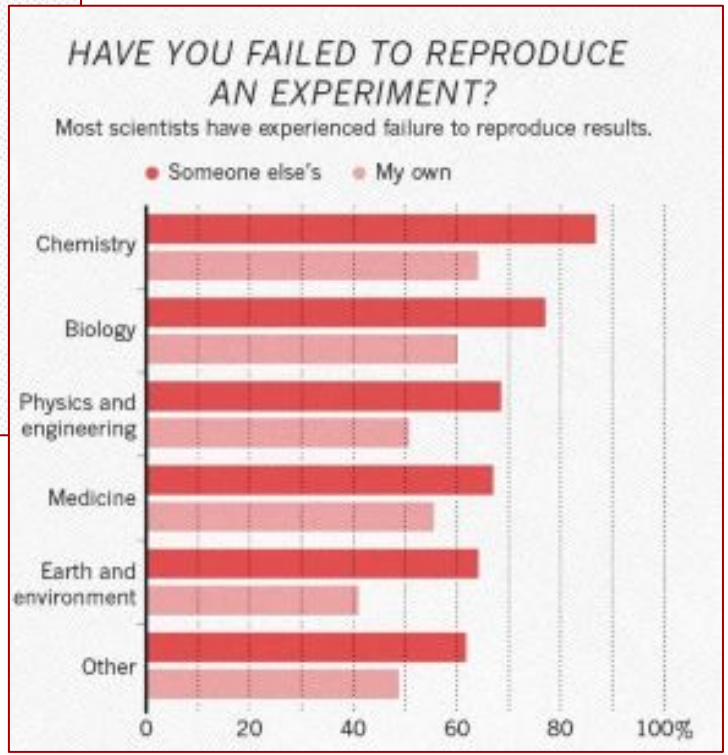
Outline for today

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



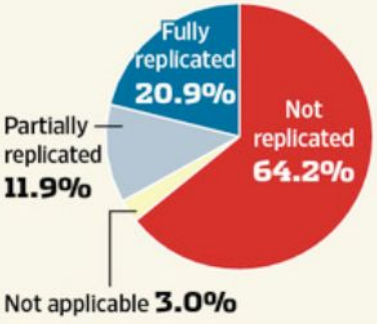
Retraction Watch

Tracking retractions as a window into the scientific process



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

Methods described in text are ambiguous and incomplete

- "**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]

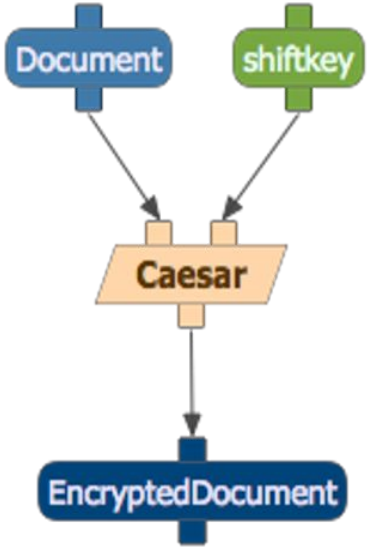
[Ince et al 2012] Ince, D., Hatton, L. & Graham-Cumming, J. The case for open computer programs. *Nature* 482, 485–488 (2012). https://doi.org/10.1038/nature10836
[Ioannidis et al 2009] Ioannidis JP, Allison DB, Ball CA, et al. Repeatability of published microarray gene expression analyses.Nat Genet. 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/

Outline for today

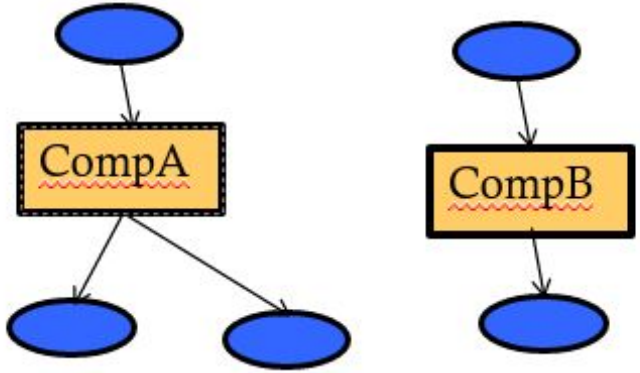
- Motivation
- **Workflows**
- Computational workflows in detail
- Workflow sketching
- Workflow systems
- The WINGS workflow system

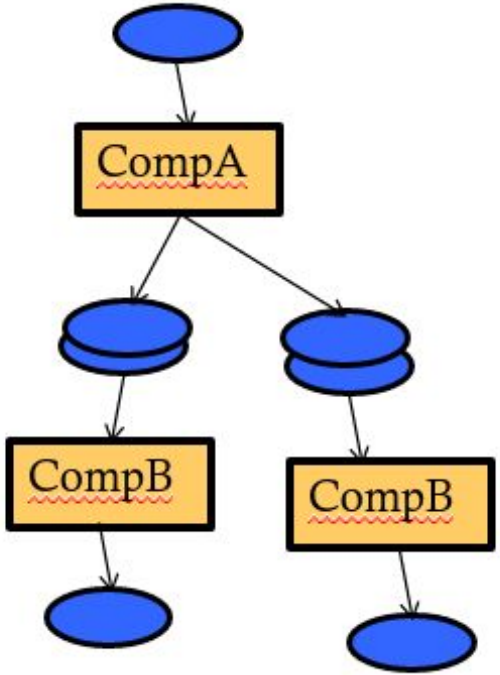
Inputs, outputs and parameters



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

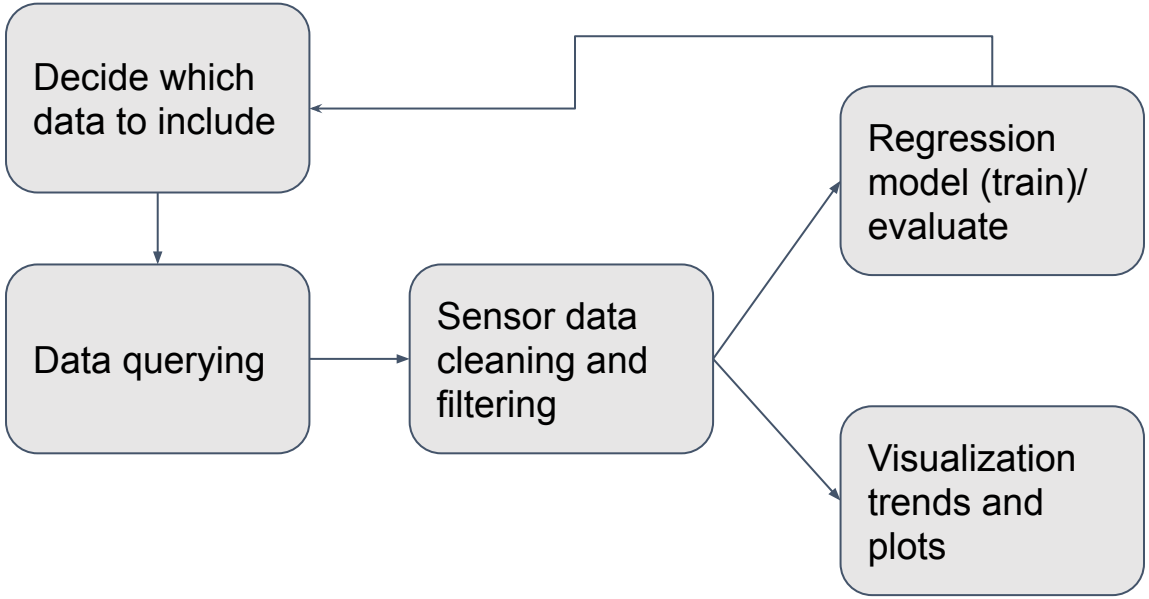
Function composition





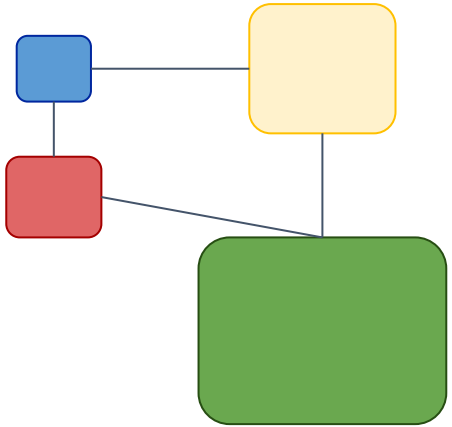
Credit: http://www.datascience4all.org/

A workflow is a composition of functions



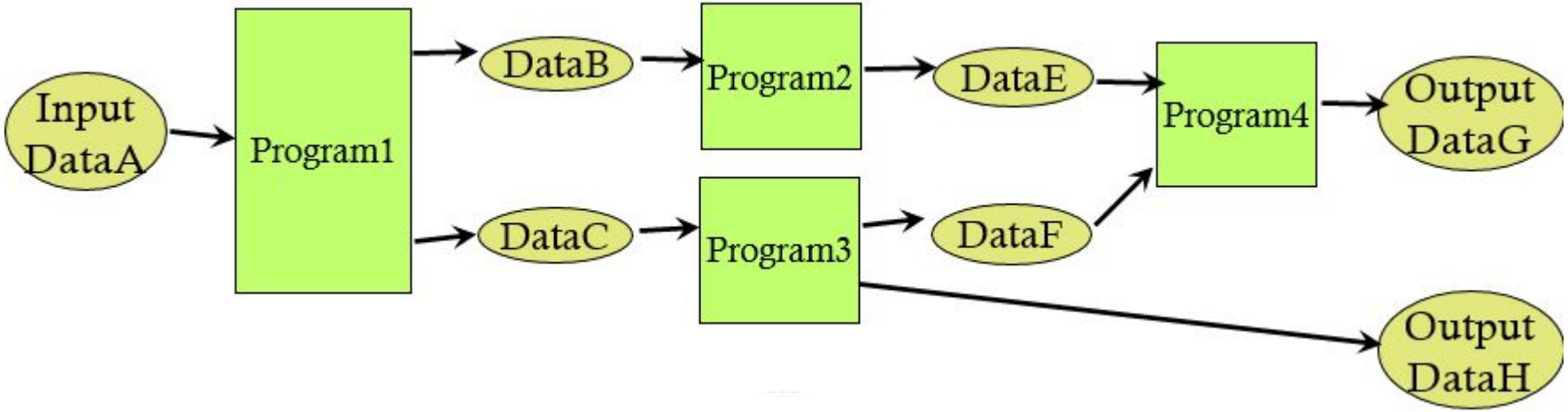
Workflow: Multi-step process

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



Computational workflow

- 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!

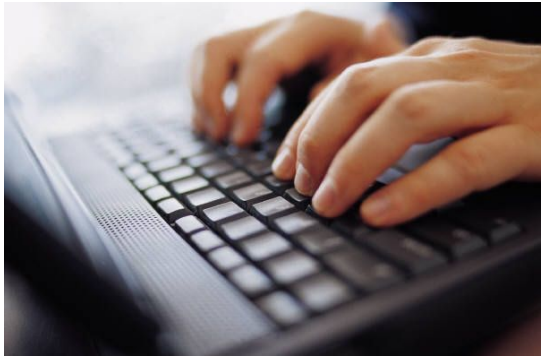


Credit: http://www.datascience4all.org/

Human activities workflow

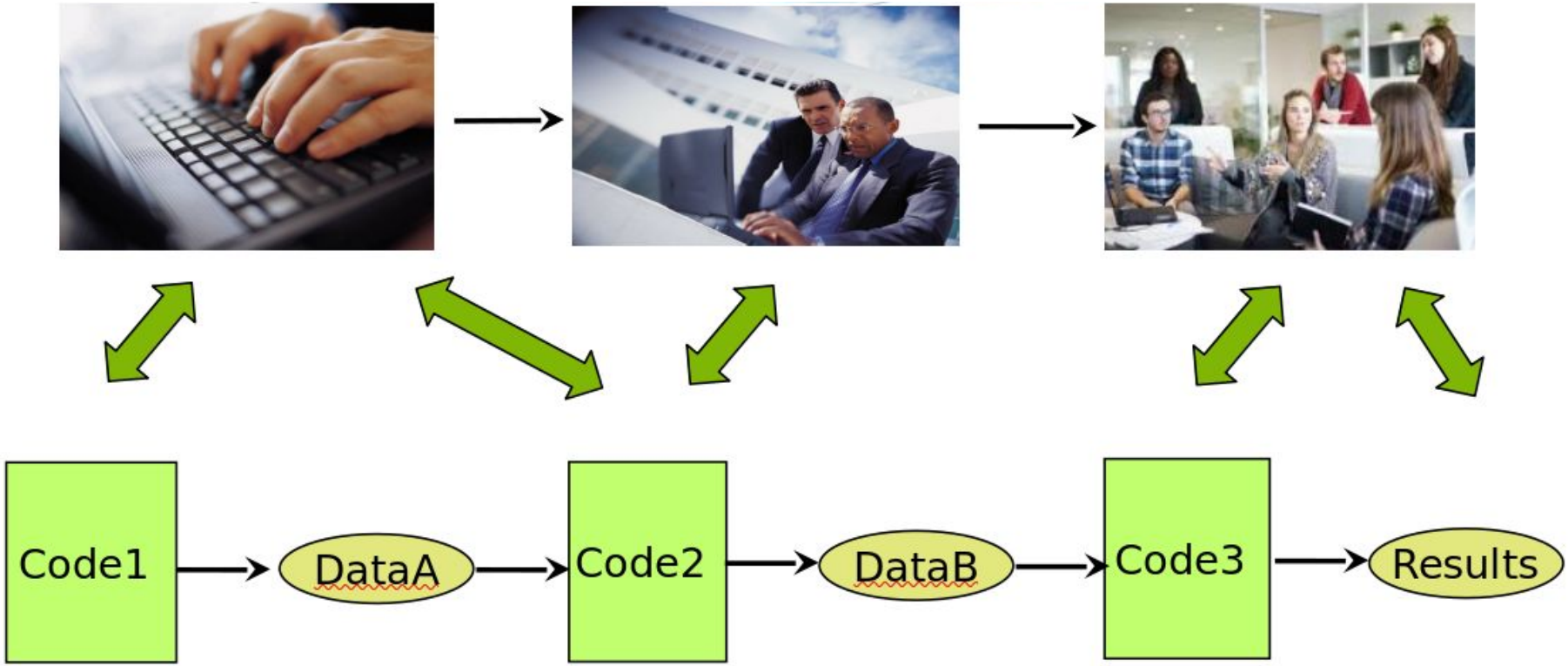
Focus is on the work being done, not the results of the work

- 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/

Hybrid workflow



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

Exercise

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

Exercise

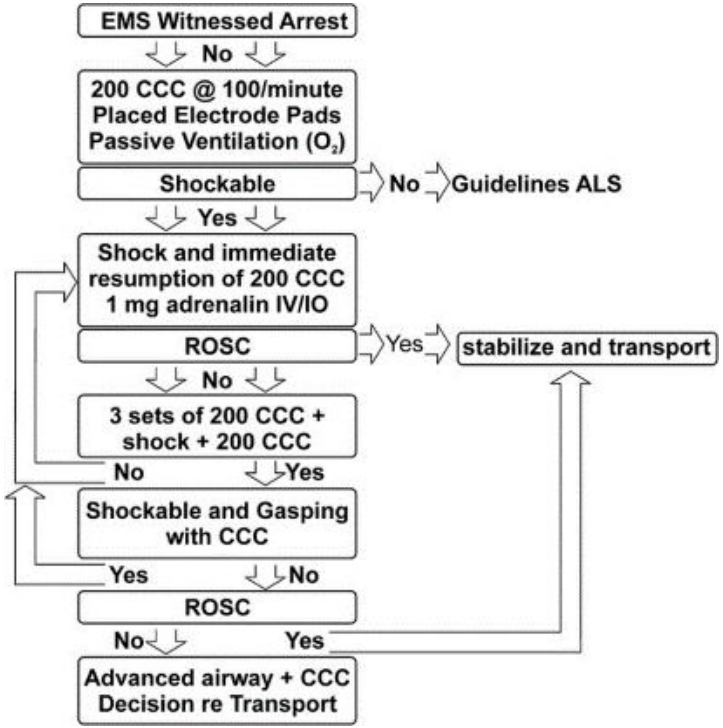
What types of workflows are the following?

- 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**

Describing a workflow from different points of view

- **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:



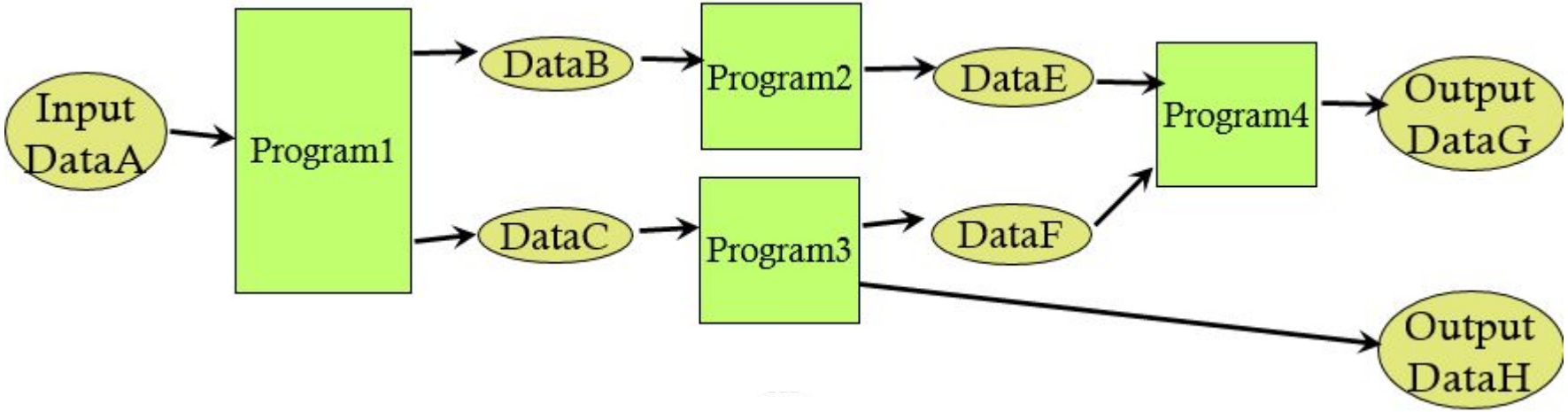
https://www.researchgate.net/figure/The-Emergency-Medical-Services-EMS-protocol-of-Cardiocerebral-Resuscitation-CCC-is_fig2_230861393

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

Outline for today

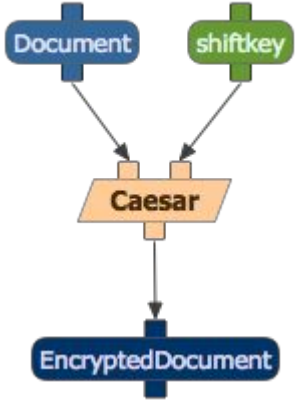
- Motivation
- Workflows
- **Computational workflows in detail**
- Workflow sketching
- Workflow systems
- The WINGS workflow system

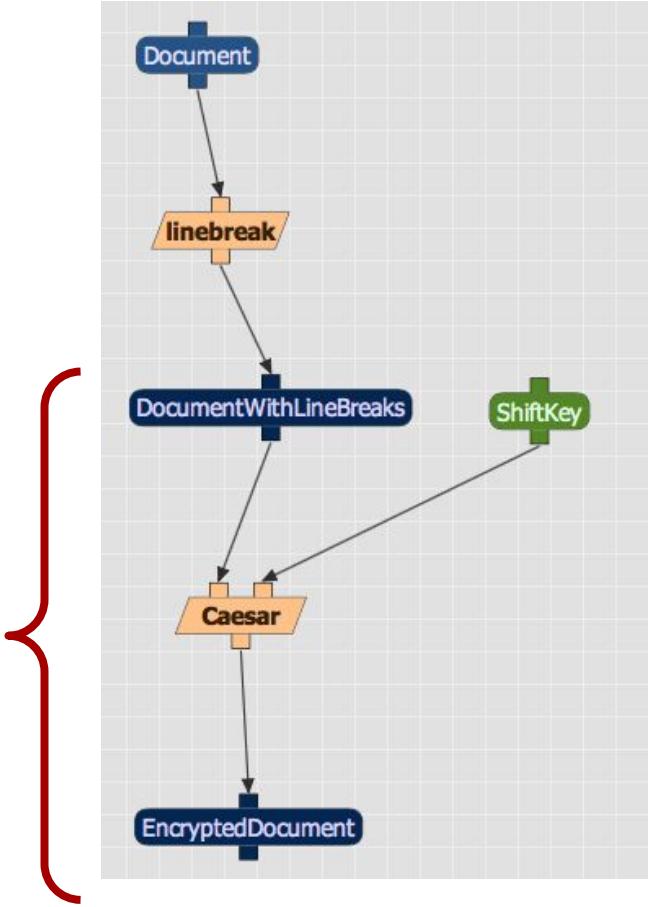
Computational Workflows in detail



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

Workflow components

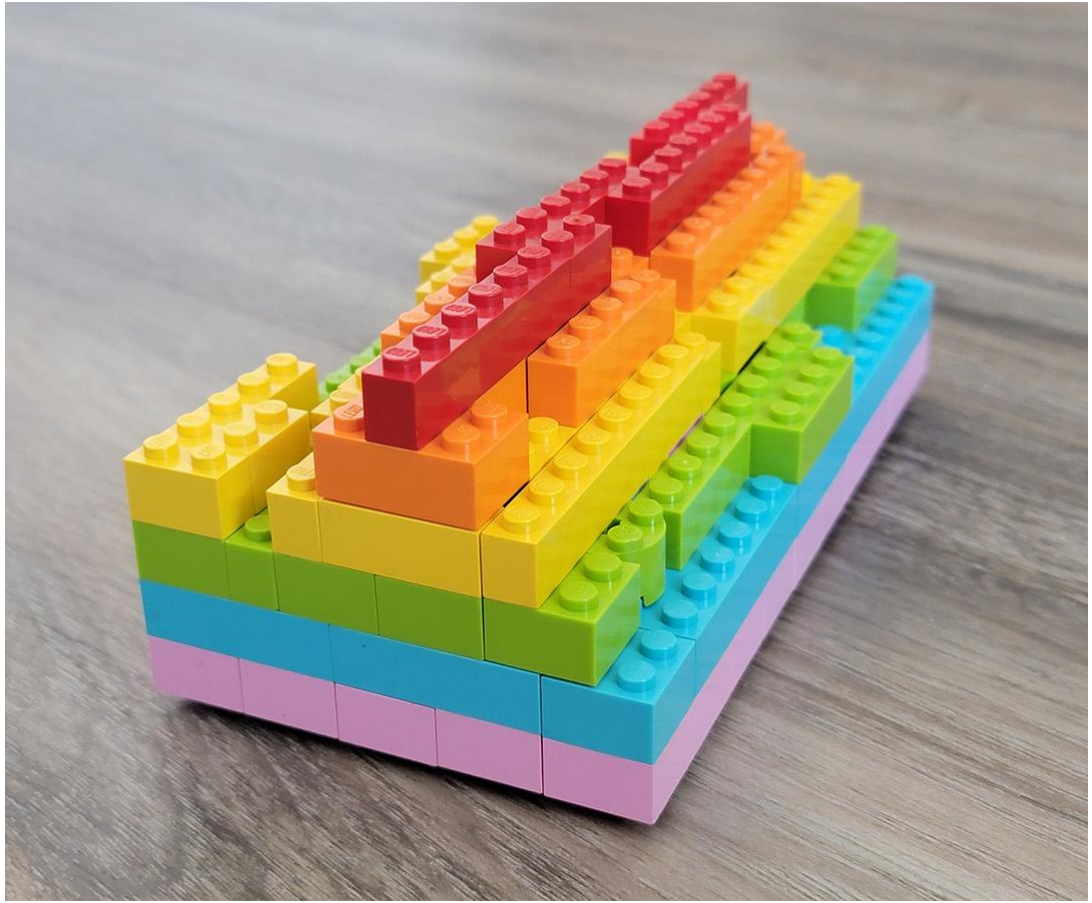




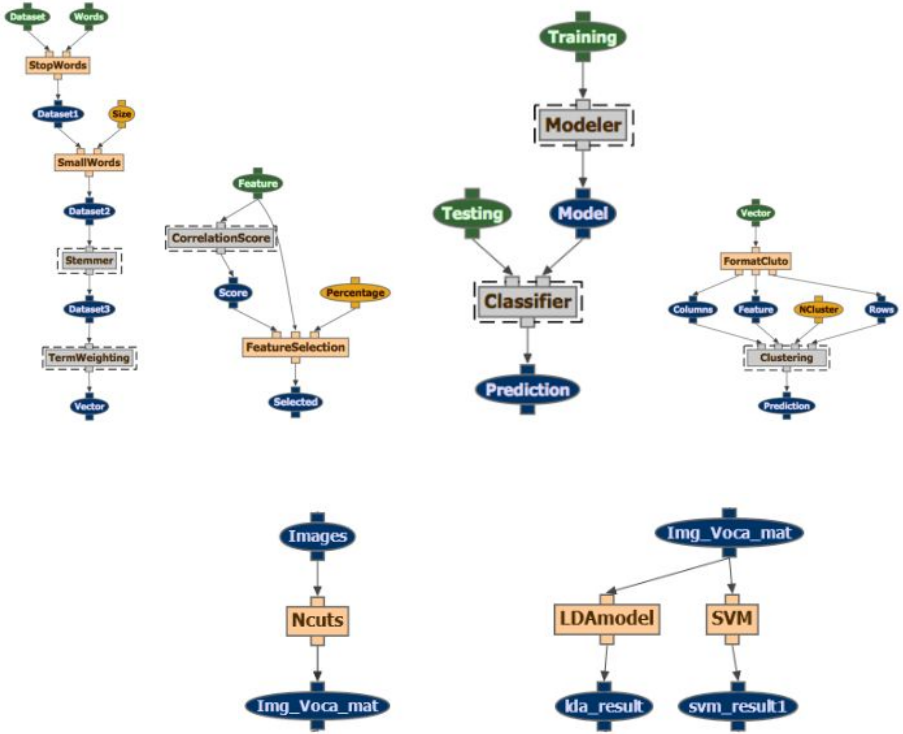
Credit: http://www.datascience4all.org/

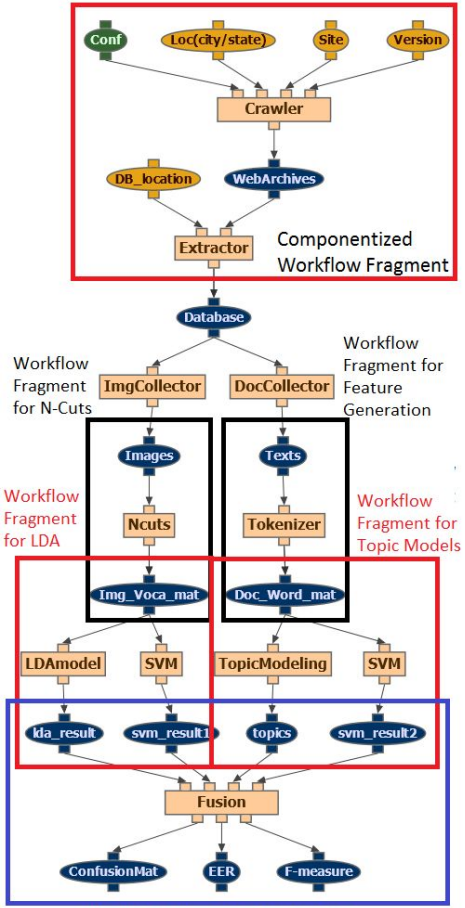
Reminds you of anything?

Building blocks!



Modular assembly

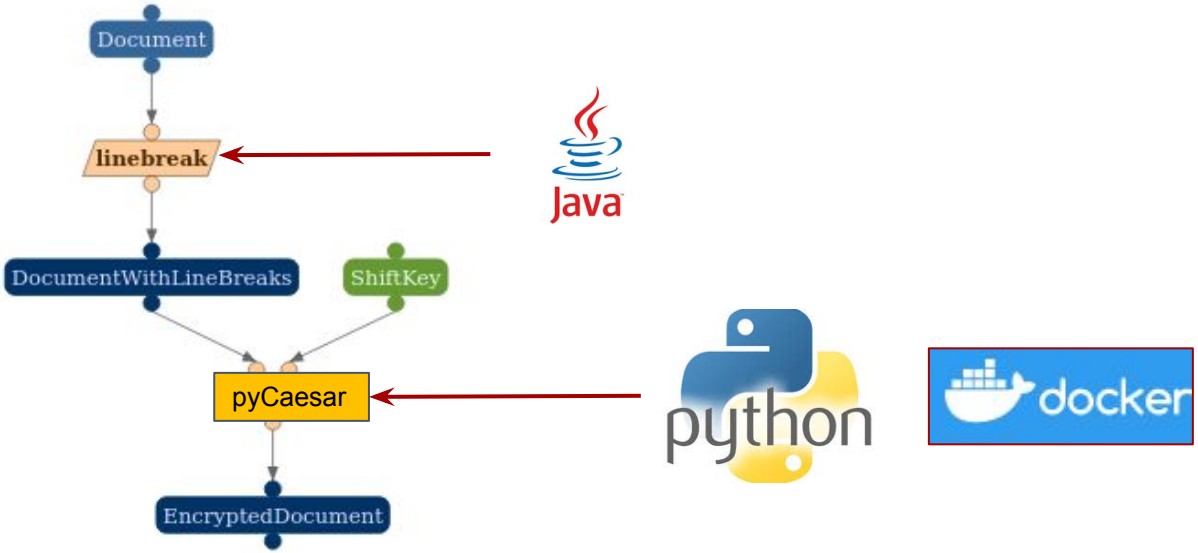




Ricky J. Sethi, Yolanda Gil, Hyunjoon Jo, and Andrew Philpot. 2013. Large-scale multimedia content analysis using scientific workflows. In Proceedings of the 21st ACM international conference on Multimedia (MM '13). Association for Computing Machinery, New York, NY, USA, 813–822. https://doi.org/10.1145/2502081.2502082

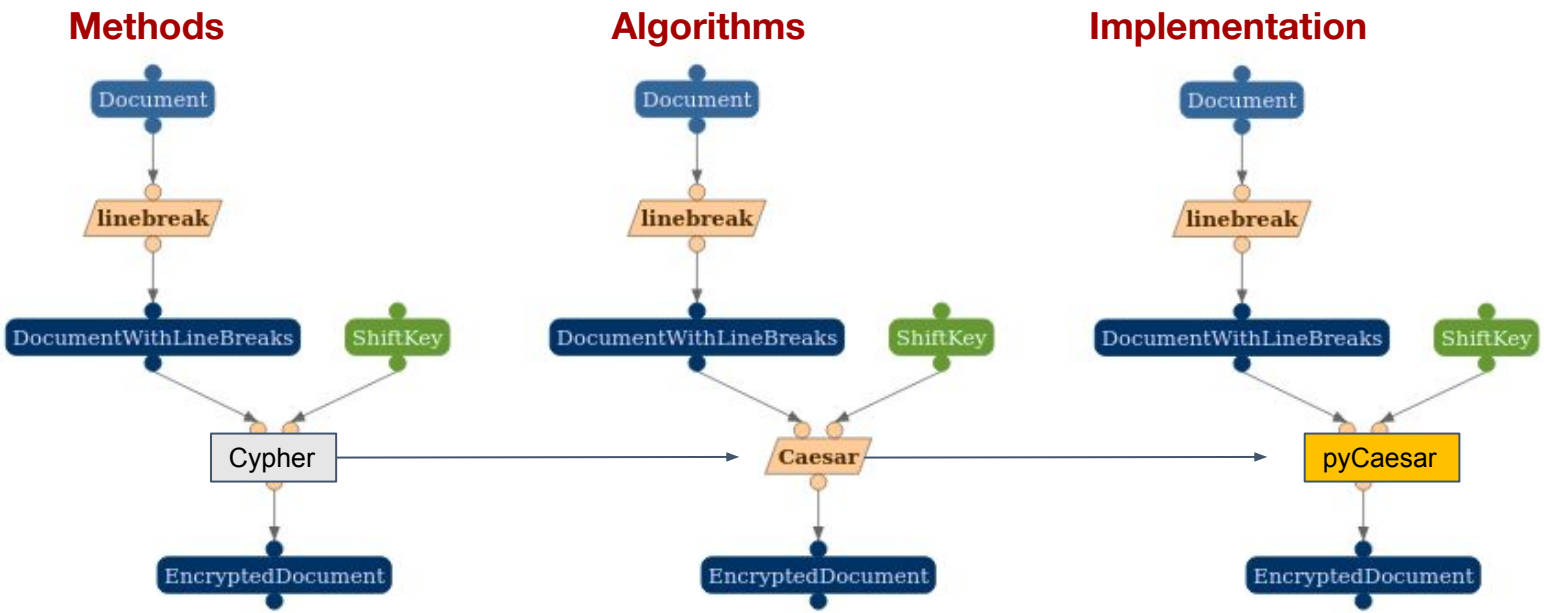
Credit: http://www.datascience4all.org/

Composing Heterogeneous Code



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

Workflows: different levels of abstraction



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

Benefits of workflows

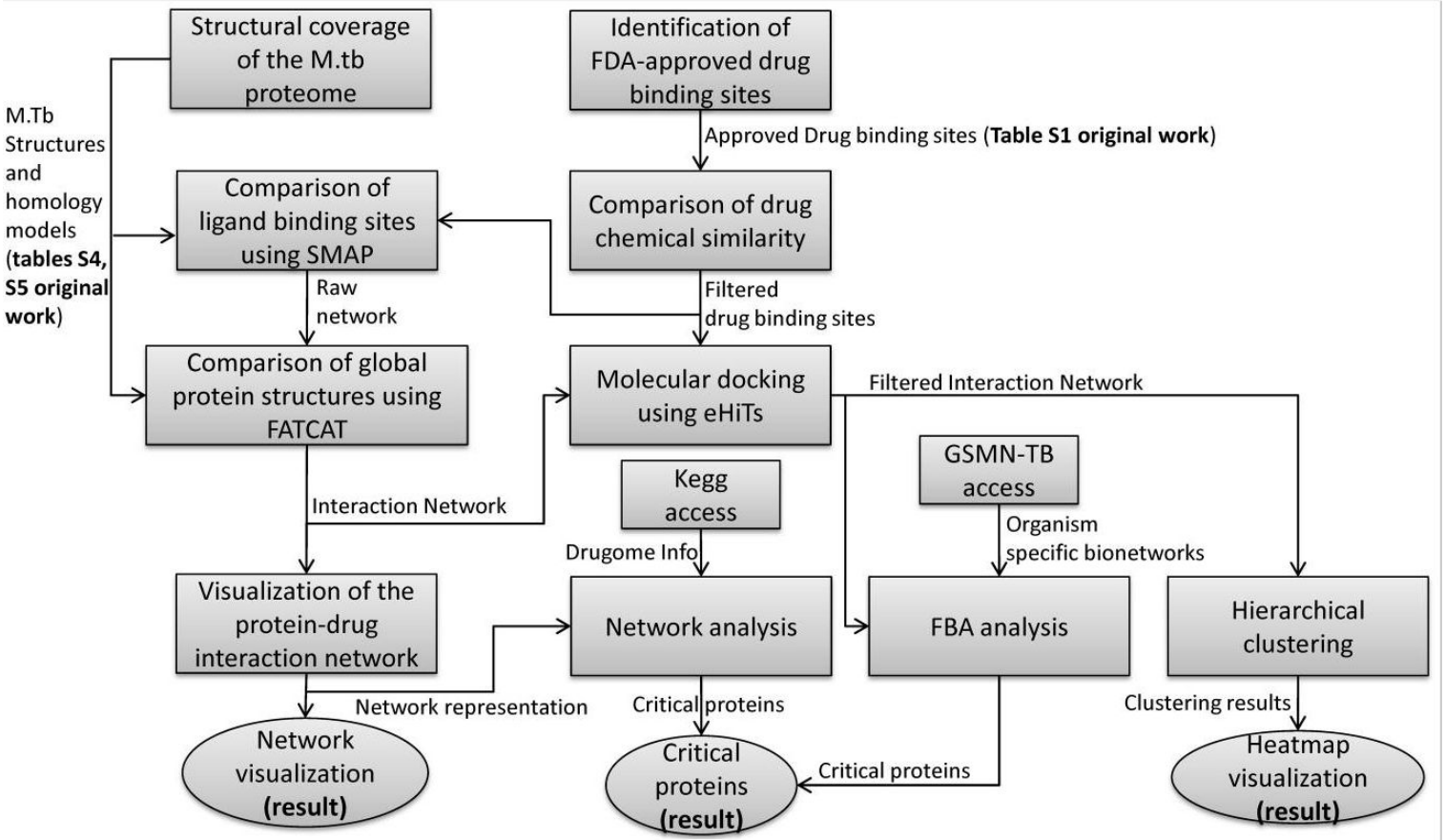
- Simple programming paradigm
- Modular assembly
- Composing heterogeneous code
- Abstraction
- Data preparation steps
- Data visualization steps
- Documenting provenance: reproducibility
- Automatic processing of multiple inputs
- Large-scale processing
- Facilitating communication across data science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

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

Outline for today

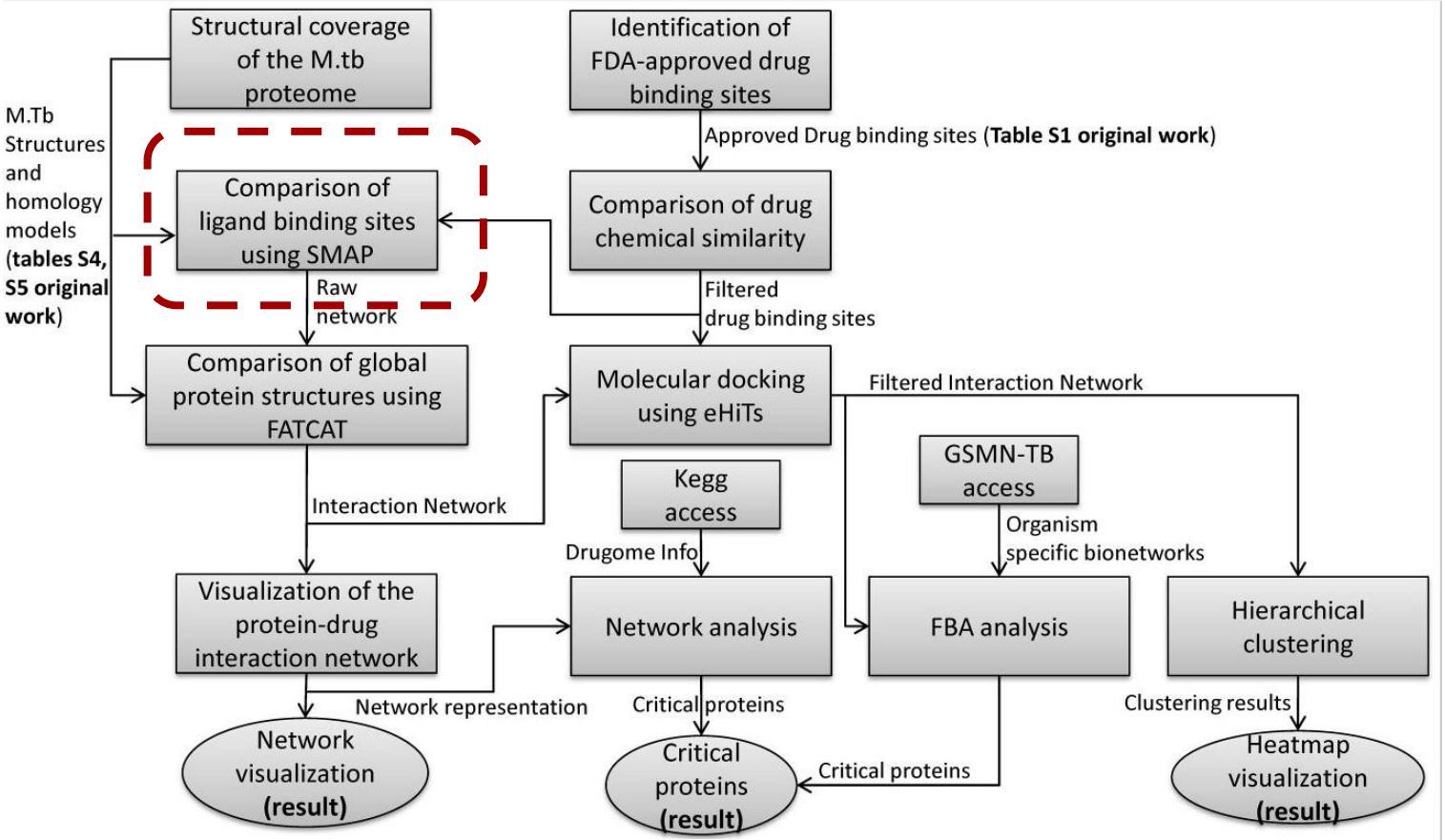
- Motivation
- Workflows
- Computational workflows in detail
- **Workflow sketching**
- Workflow systems
- The WINGS workflow system

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



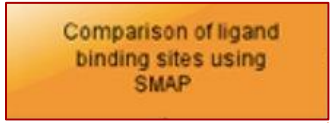
Garijo D, Kinnings S, Xie L, Xie L, Zhang Y, Bourne PE, et al. (2013) Quantifying Reproducibility in Computational Biology: The Case of the Tuberculosis Drugome. PLoS ONE 8(11): e80278. https://doi.org/10.1371/journal.pone.0080278

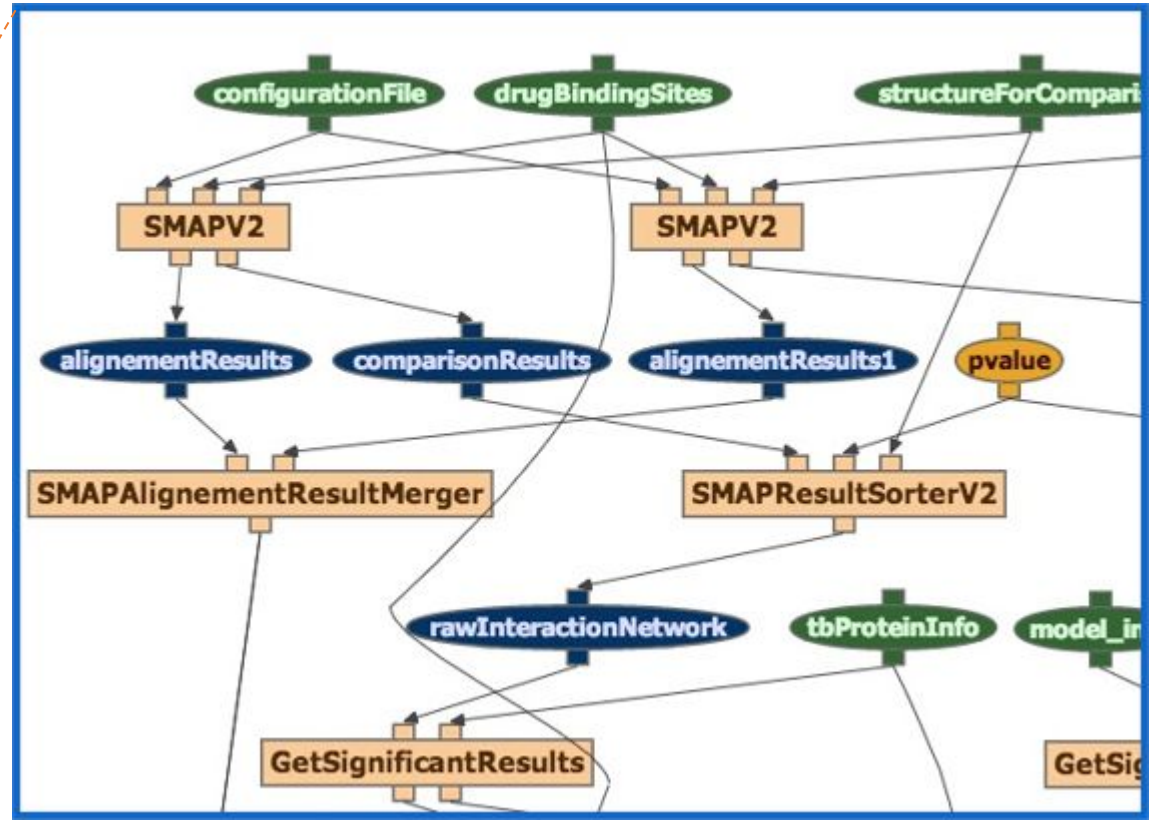
What the paper says versus what the experiment does (1)



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)



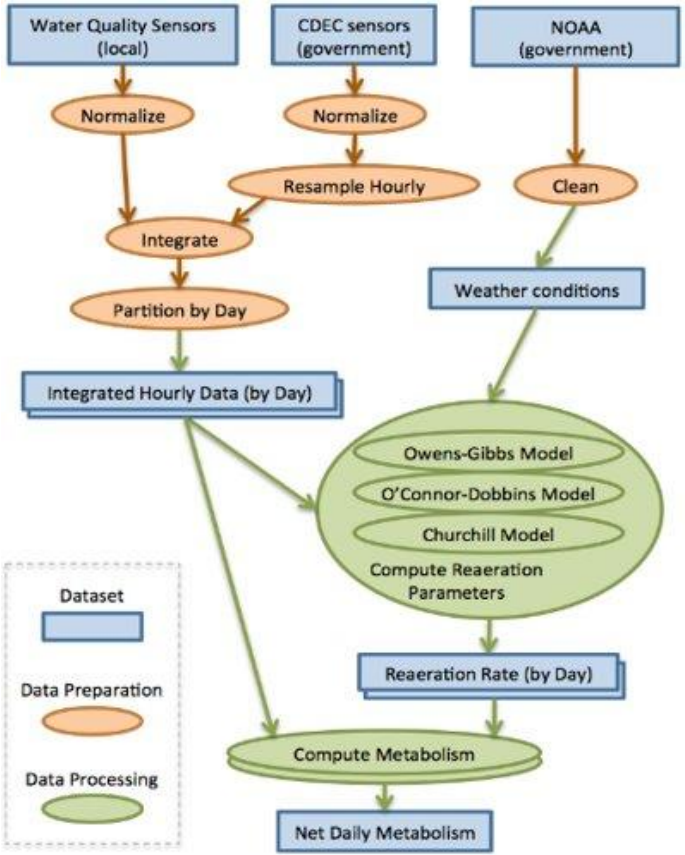


Garijo D, Kinnings S, Xie L, Xie L, Zhang Y, Bourne PE, et al. (2013) Quantifying Reproducibility in Computational Biology: The Case of the Tuberculosis Drugome. PLoS ONE 8(11): e80278. https://doi.org/10.1371/journal.pone.0080278

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

How to sketch a workflow

1. Compile the command line invocation to all your codes
   - Input data, parameters, configuration files
   - 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
   - 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. […]

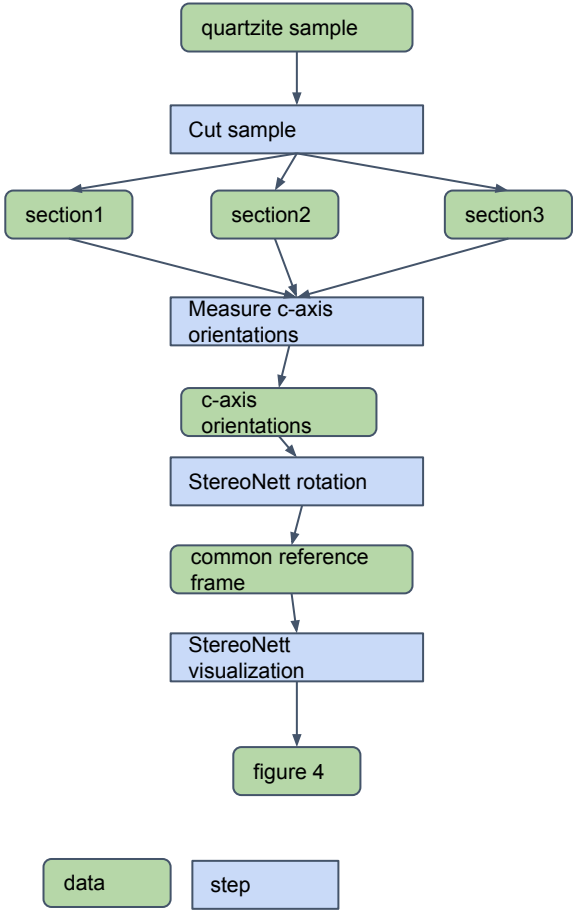
Understanding kinematic data from the Hellerman thrust zone, Jade Silverstein

Sketching a workflow: solution

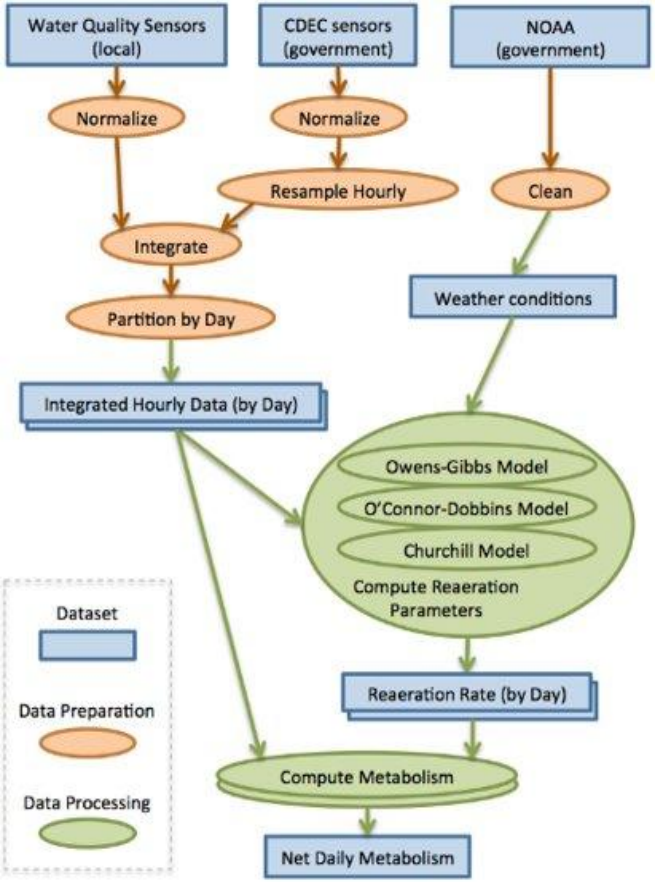
Draw a step-by-step process of the workflow you understand from the following text:

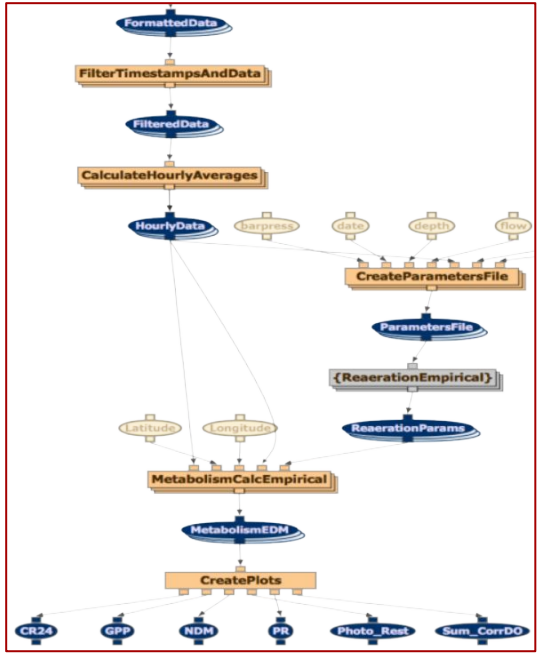
[…] We took a quartzite sample from the Hellerman thrust zone, and cut 3 thin sections. We measured c-axis orientations using a petrographic microscope. We rotated to a common reference frame using Duyster's StereoNett program. We plotted the data on lower hemisphere, equal area projections using Duyster's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

Sketching a workflow

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

Some volunteers to discuss the results!

Outline for today

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

Workflow systems

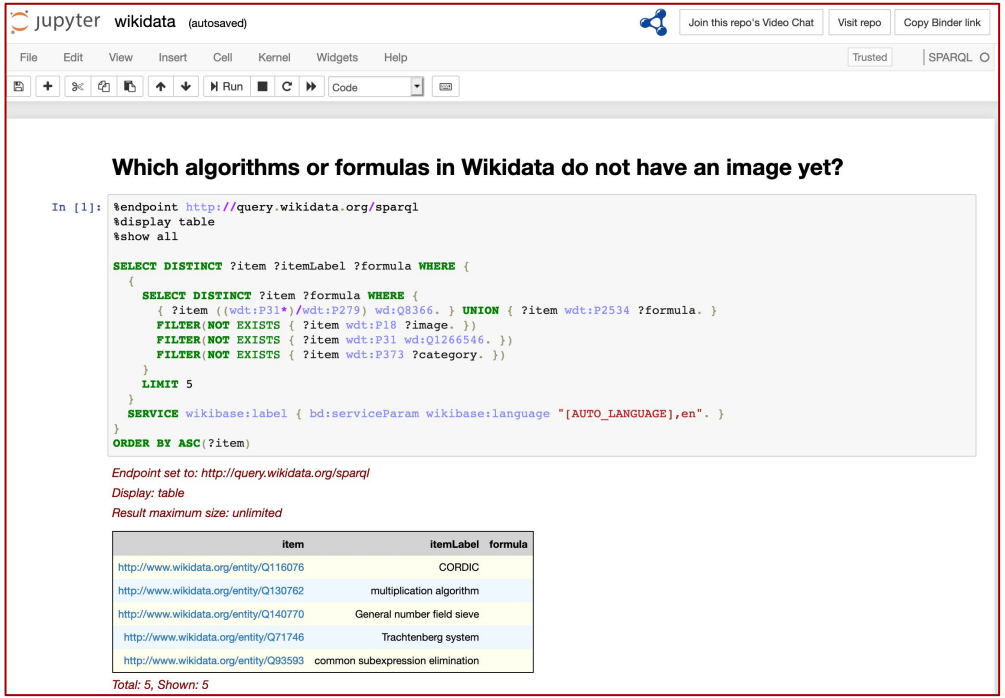
- Many choices
  - Academic prototypes
  - Operational open source
  - Commercial

- Each has different capabilitie
  - Workflow validation
  - Scalable computations
  - Domain functions

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

Computational notebooks vs workflow systems

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



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

Workflows versus notebooks

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/

Outline for today

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

WINGS workflows (live demo)

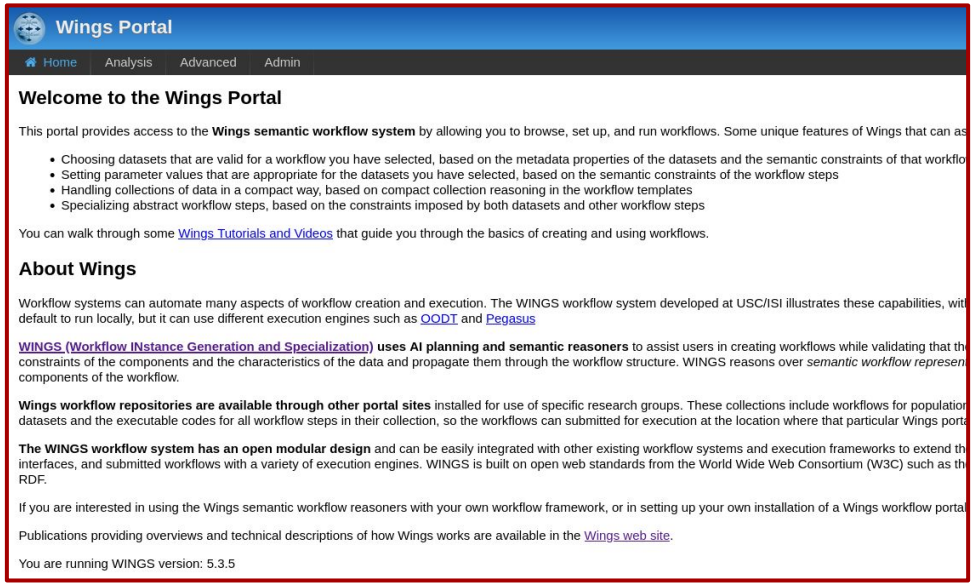
https://www.wings-workflows.org/tutorial/tutorial.html

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

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

http://localhost:8080/wings-portal

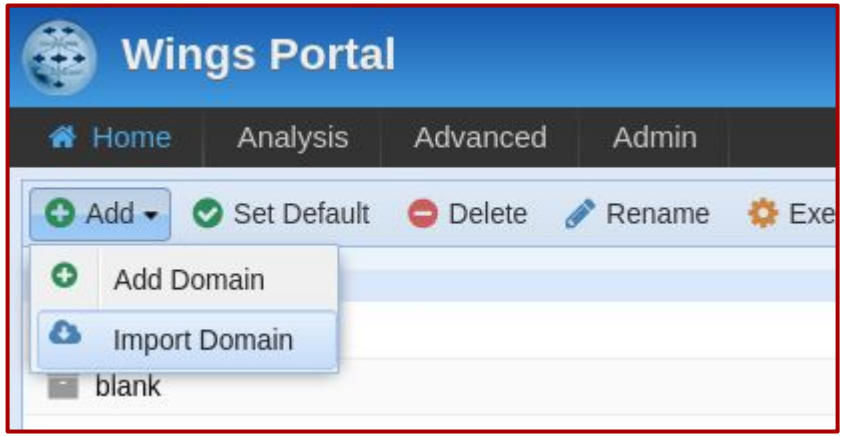
user/pass: admin/admin123

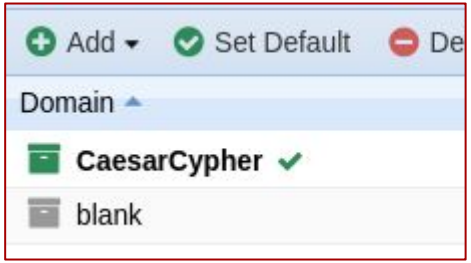


Start by importing a domain:

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

Advanced -> Manage domain -> Import domain





WINGS workflows (live demo)

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)

Next TO DOs

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)

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