Theory & Practice of Data Cleaning

Workflow Automation and Provenance

Reproducibility Crisis (hitting the airwaves)

- e.g. Science, Economist, Nature (via SciAM):
  - https://www.scientificamerican.com/video/is-there-a-reproducibility-crisis-in-science/
- BBC4 report :
  - Causes: culture that rewards novel, eye-catching results, i.e.,
    - **inappropriate statistics**
    - **selective reporting**
    - **hyping**

Computational Reproducibility

- Different sciences have different reproducibility crises …
- Our focus: **Computational Reproducibility**
  - **Scientific workflows**,
  - … and **scripts**: R, Matlab, Python, ..

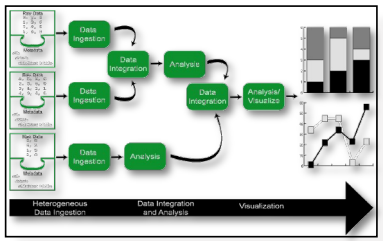
- ***How to facilitate reproducibility for computational and data scientists?***
  - **Workflow Automation** (workflow systems, scripts)
  - **Transparency**
  - **Provenance**

Reproducibility Studies

- **Successful** reproducibility study:
  - **increases trust** in prior study ☺
  - … but **no surprises** ☹
- **Failed** reproducibility study :
  - **decreases trust** (or *falsifies*) prior study ☹
  - … but **surprising** failure yields **new info/knowledge** ☺
- Learning from failures!
  - Not really a new, revolutionary idea..
  - What is a positive vs negative result anyways?
  - … *fail early, fail often* …

Scientific Workflows: ASAP

- **A**utomation
  - wfs to **automate** computational aspects of science



- **S**caling (exploit and optimize ***machine*** *cycles*)
  - wfs should make use of **parallel compute resources**
  - wfs should be able handle **large data**

Kepler

swift

- **A**bstraction, Evolution, Reuse *(**human** cycles)*
  - wfs should be easy to **(re-)use, evolve, share**

Taverna

ASKALON

VisTrails

TRIANA

- **P**rovenance
  - wfs should capture **processing history**, **data lineage**

➔ traceable data- and wf-evolution

➔ **Reproducible Science**

pegasus

Trident Workbench

Es war einmal …

Essential Functions of a Scientific Workflow System

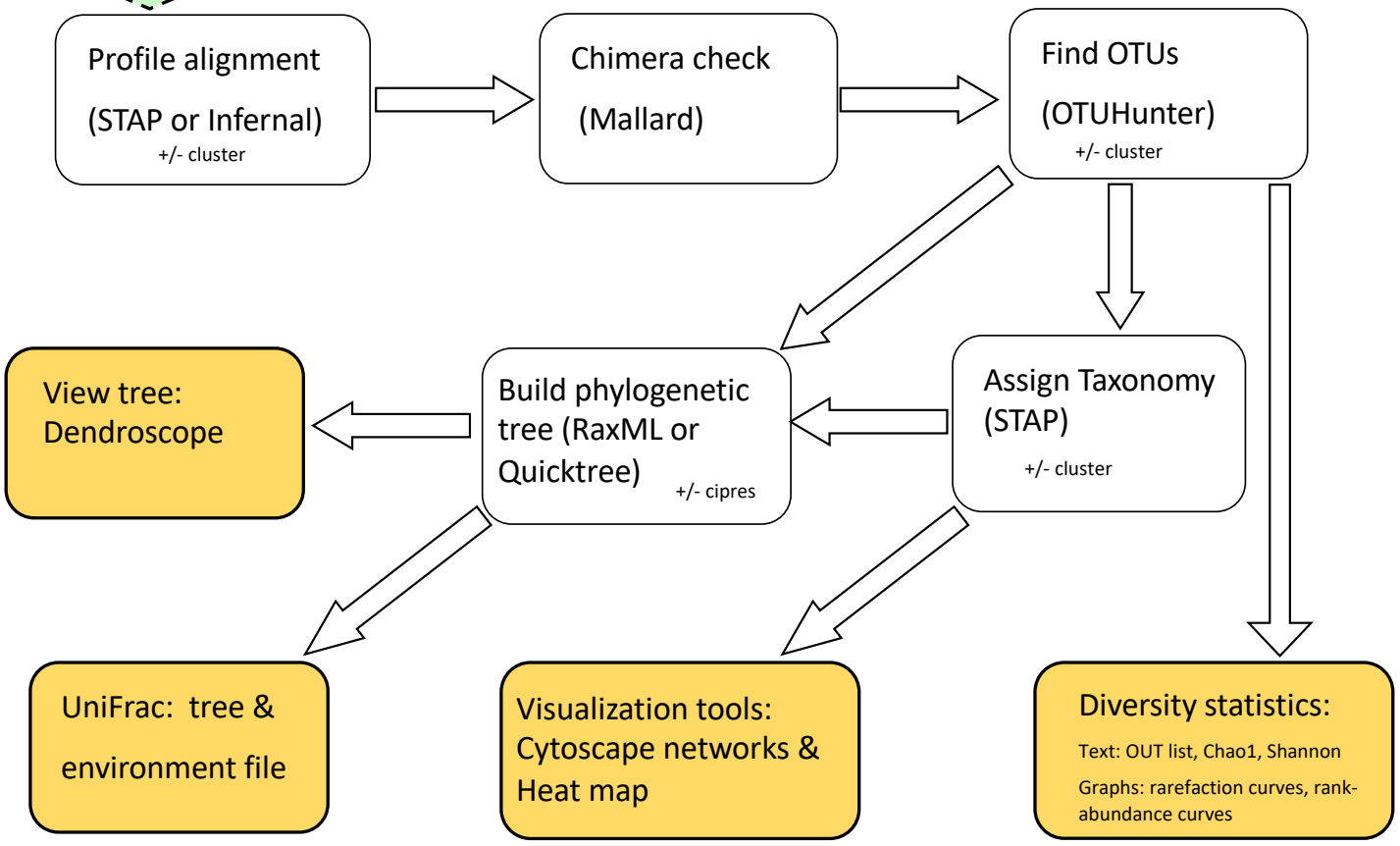
1. **Automate** programs and services scientists already use.
2. **Schedule** invocations of programs and services correctly and efficiently – **in parallel** where possible.
3. **Manage dataflow** to, from, and between programs and services.
4. **Enable scientists** (not just developers) to **author** or **modify** workflows easily.
5. **Predict** what a workflow will do when executed: ***prospective* provenance**.
6. **Record** what happened during workflow execution: ***retrospective* provenance**.
7. **Reveal and query provenance** – how workflow products were derived from inputs via programs and services.
8. **Organize** intermediate and final **data products** as desired by users.
9. Enable scientists to **version**, **share** and **publish** their workflows.
10. **Empower scientists** who wish to **automate additional programs and services themselves.**

These functions (not just dataflow & actors) distinguish ***scientific workflow automation*** from general (scientific) software development.

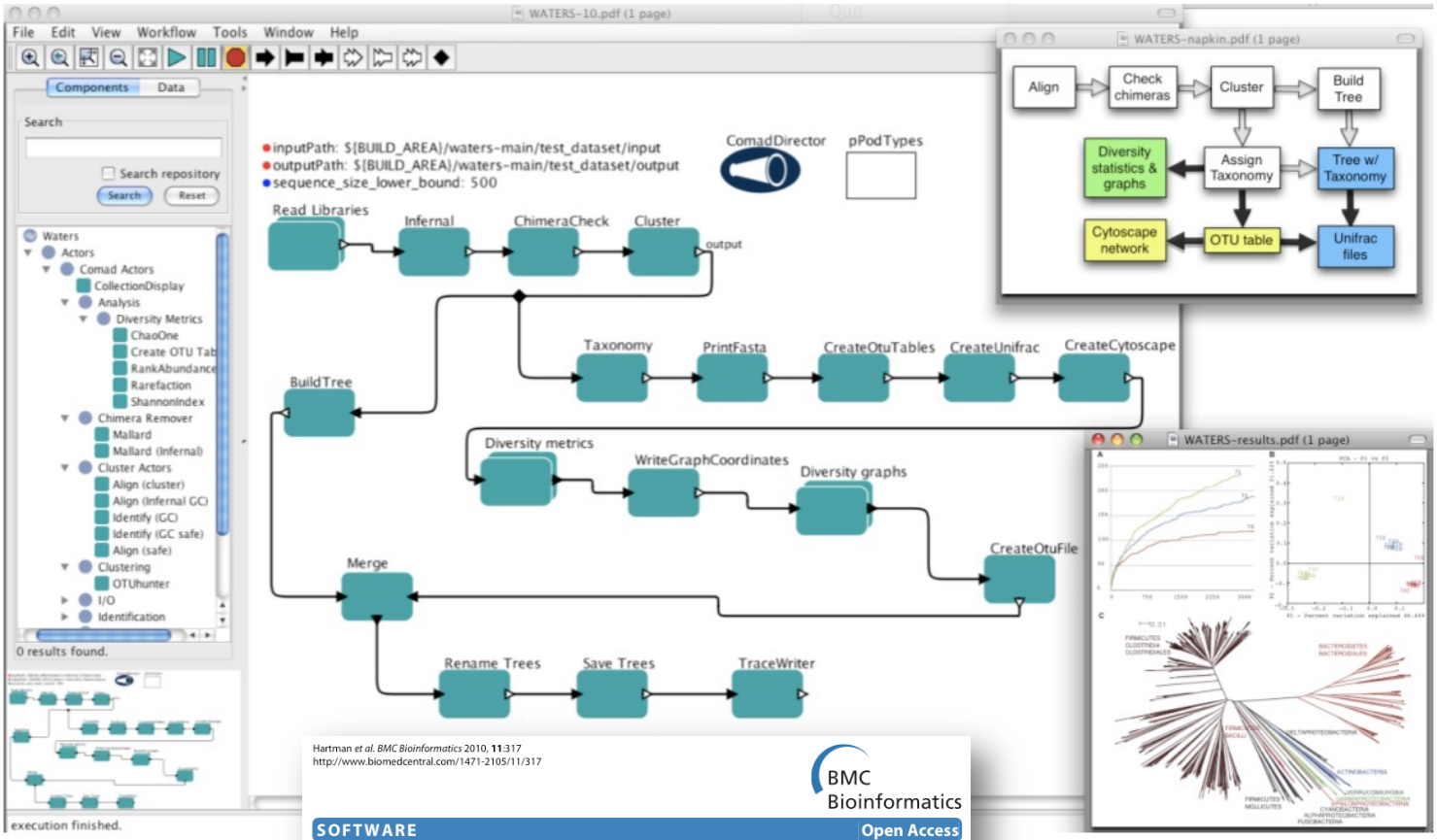
Src: Tim McPhillips

WATERS: Workflow for Alignment, Taxonomy, Ecology of Ribosomal Sequences

(Amber Hartman; Eisen Lab; UC Davis)



Executable WATERS Workflow in Kepler



Hartman *et al. BMC Bioinformatics* 2010, **11**:317
http://www.biomedcentral.com/1471-2105/11/317

BMC Bioinformatics

SOFTWARE Open Access

Introducing W.A.T.E.R.S.: a Workflow for the Alignment, Taxonomy, and Ecology of Ribosomal Sequences

Amber L Hartman[†1,3], Sean Riddle[†2], Timothy McPhillips[2], Bertram Ludäscher[2] and Jonathan A Eisen[*1]

Example Bioinformatics Workflow:

Motif-Catcher

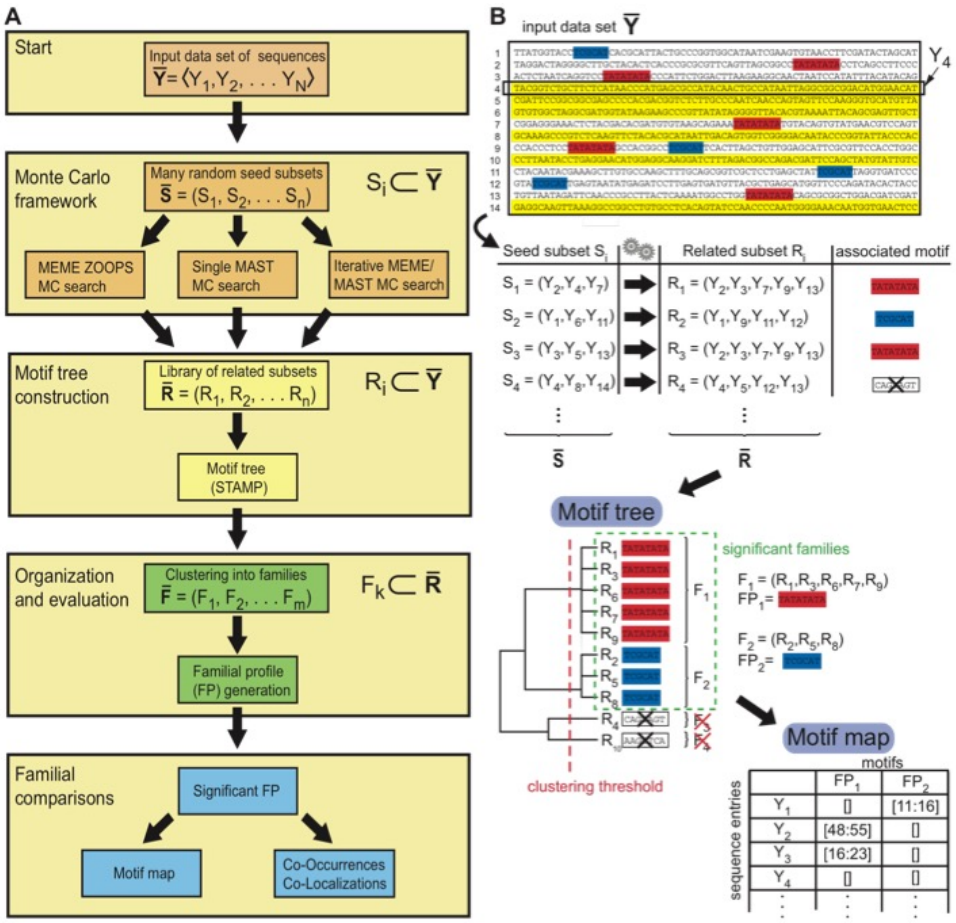
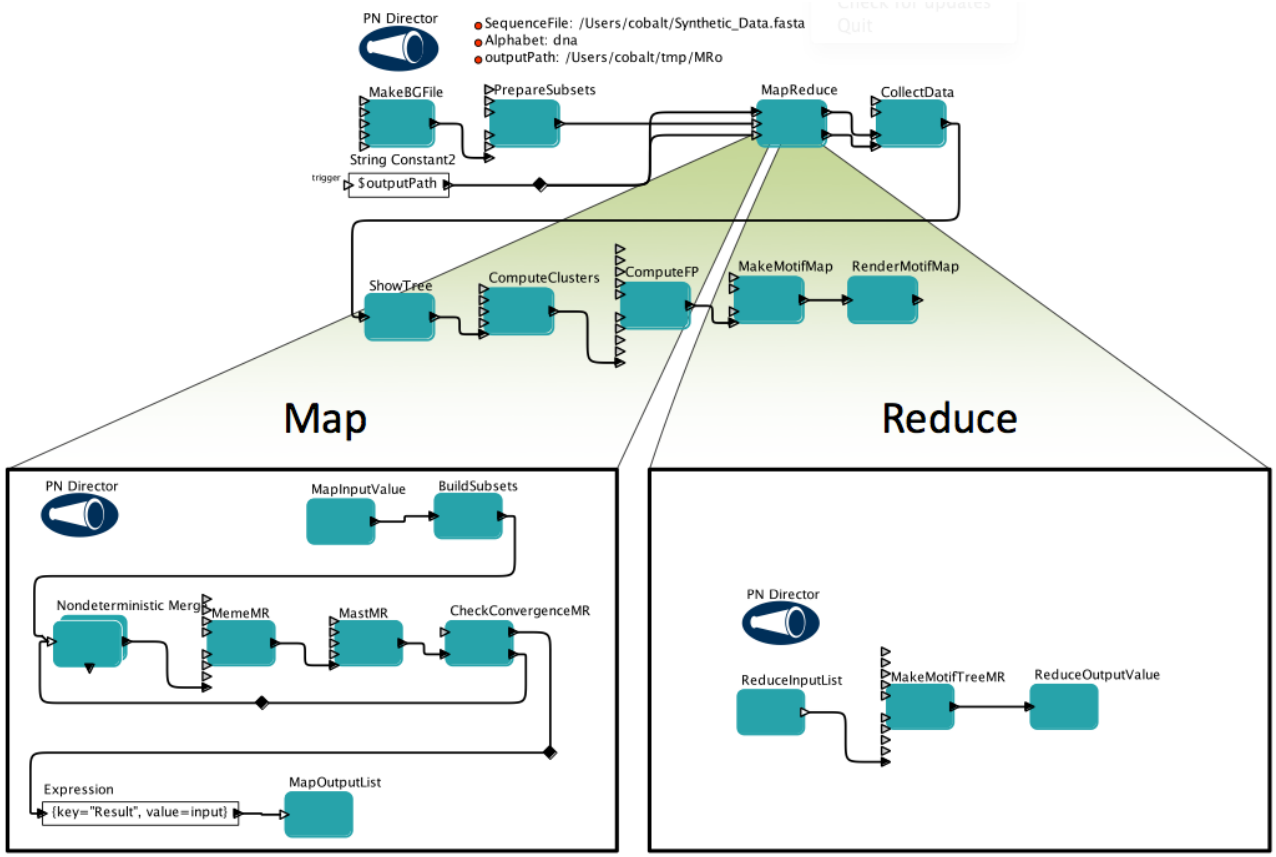


Figure 1: Concept of Monte-Carlo based detection and interpretation of motifs. A) Abstract description of MotifCatcher process. B) Examples illustrating the process with sample data.

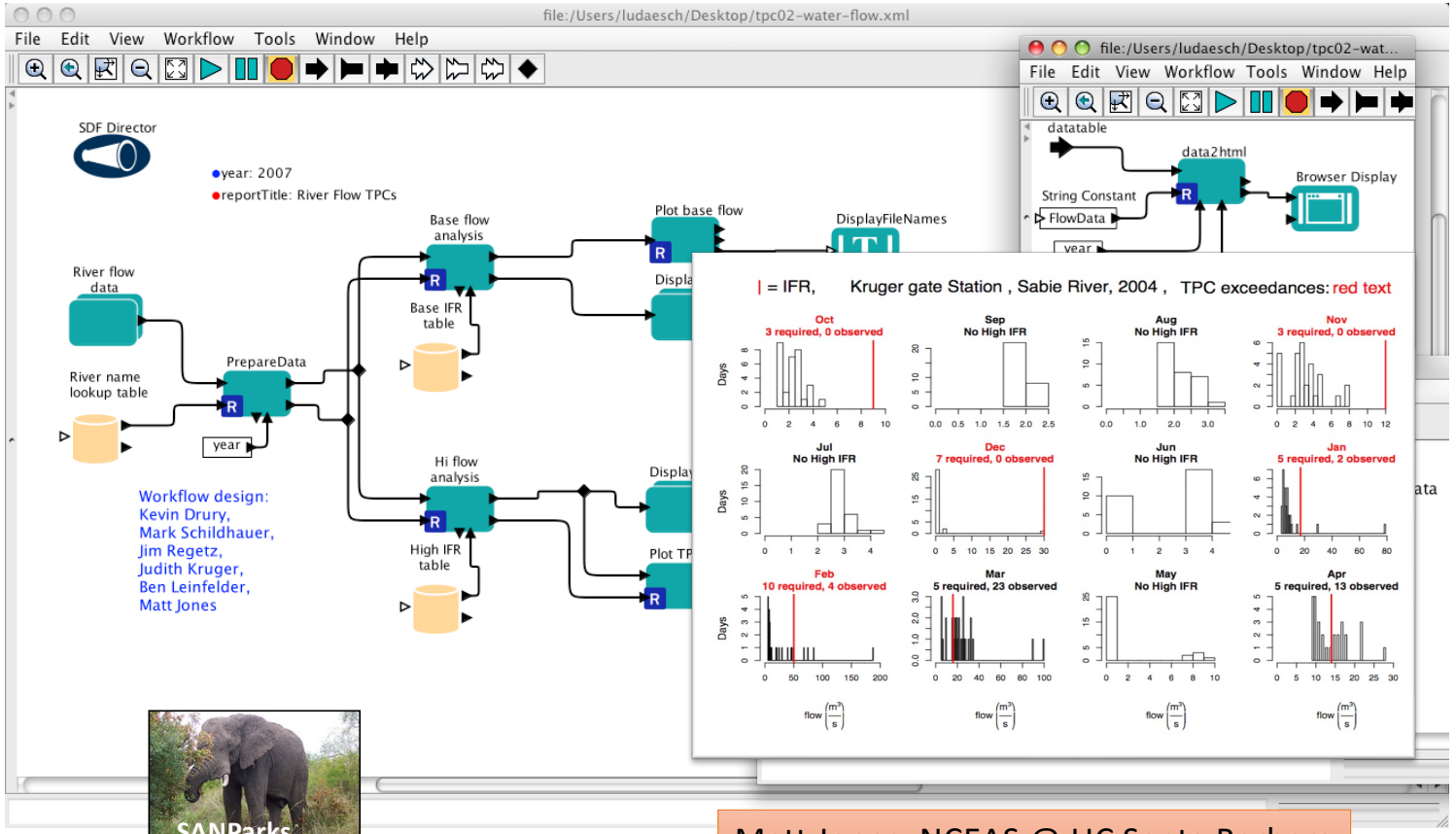
Marc Facciotti *et al.*
UC Davis Genome Center

Motif-Catcher workflow, implemented in Kepler



S Köhler et al. Improved Motif Detection in Large Sequence Sets with Random Sampling in a Kepler workflow, *ICCS-WS*, 2012

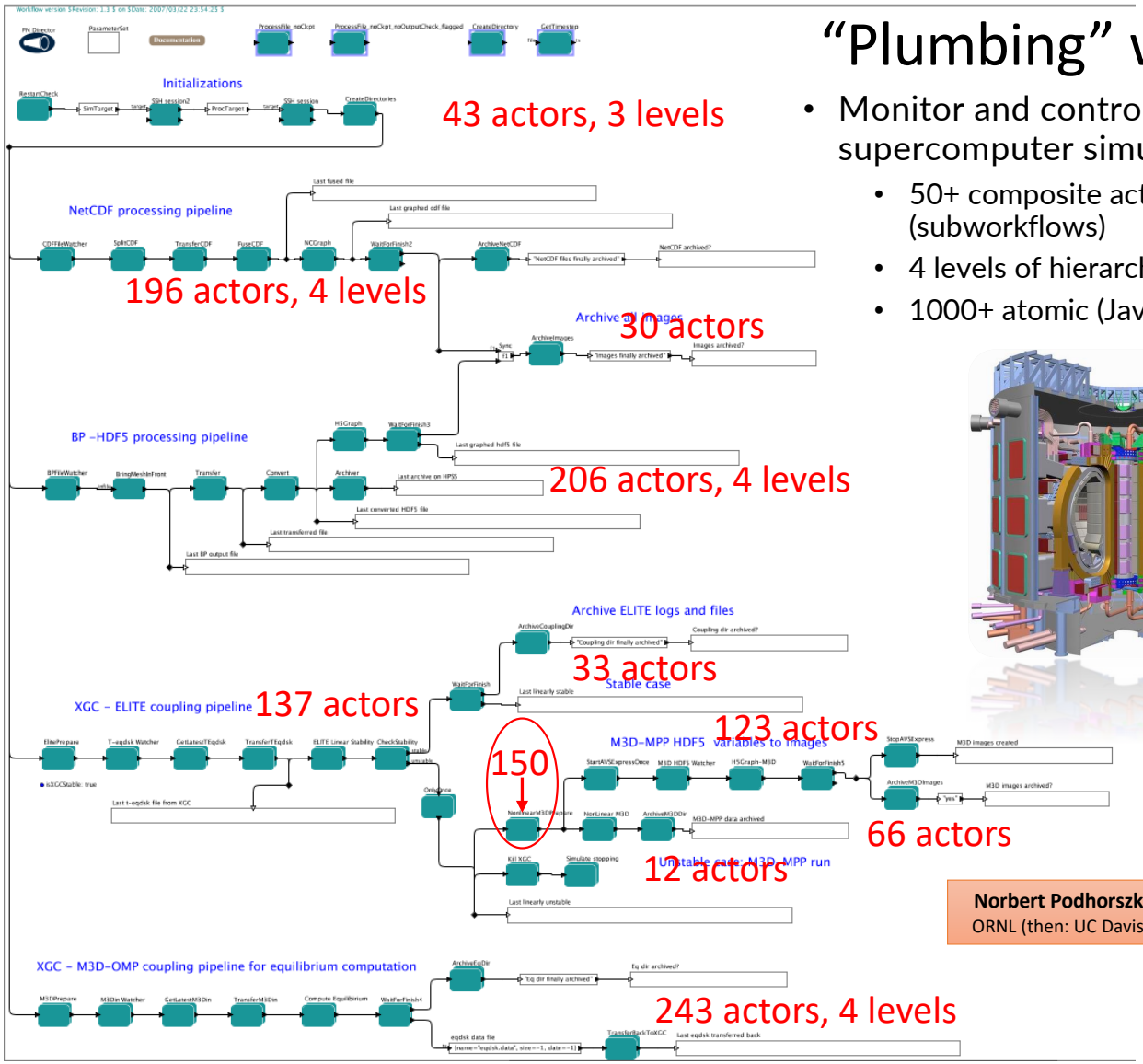
Kepler Workflows & Decision Making
(Kruger Natl. Park, South Africa)



Matt Jones, NCEAS @ UC Santa Barbara

“Plumbing” workflow

- Monitor and control supercomputer simulations
  - 50+ composite actors (subworkflows)
  - 4 levels of hierarchy
  - 1000+ atomic (Java) actors

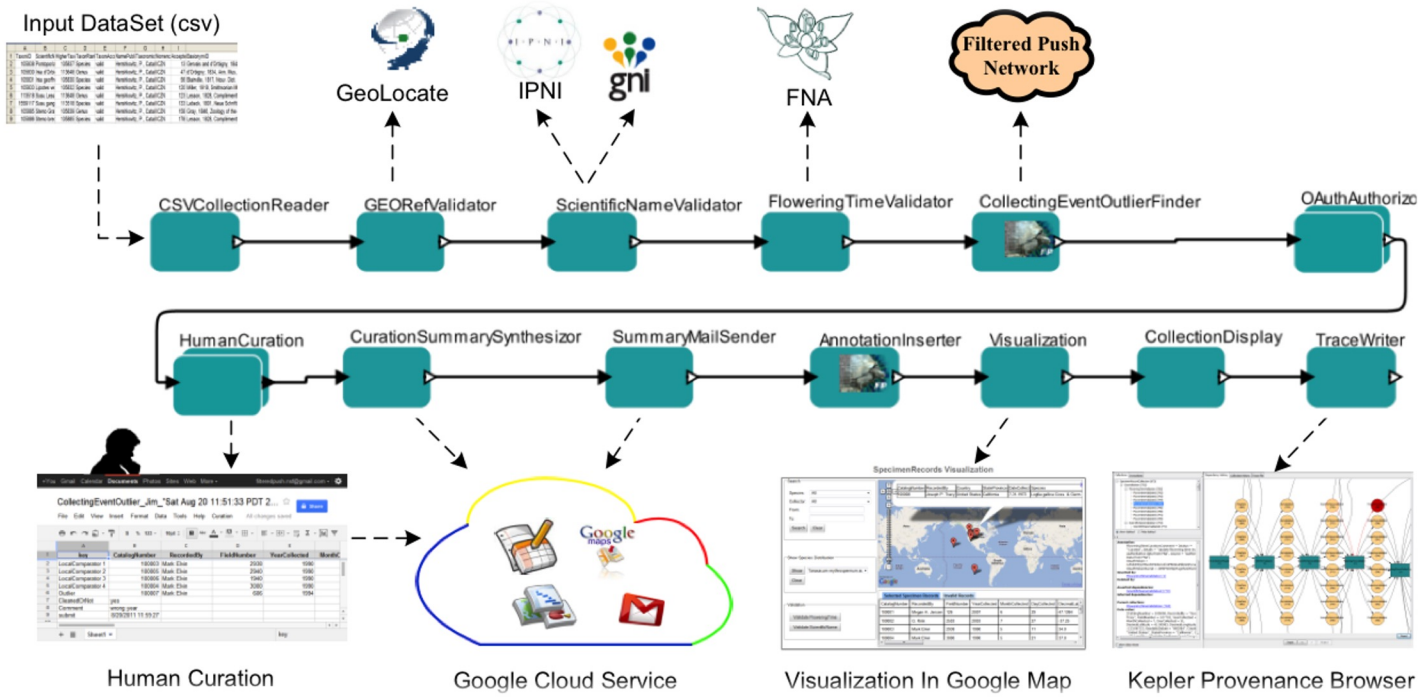


Norbert Podhorszki
ORNL (then: UC Davis)

Center for Plasma Edge Simulation

Data Curation Workflows

(*Filtered-Push … Kepler … Kurator projects*)



HARVARD UNIVERSITY HERBARIA

MUSEUM OF COMPARATIVE ZOOLOGY
HARVARD UNIVERSITY

UCDAVIS Genome center

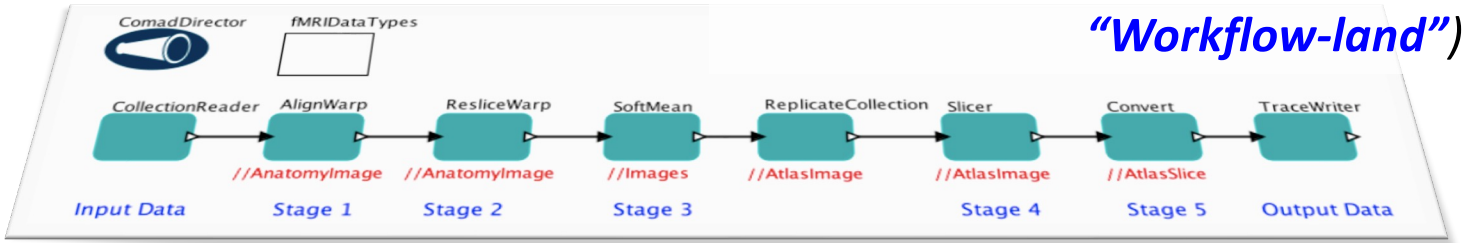
ILLINOIS
1867
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Provenance ⇔ Workflows

Workflow Modeling & Design

(a.k.a. ***prospective*** *provenance*

"Workflow-land")



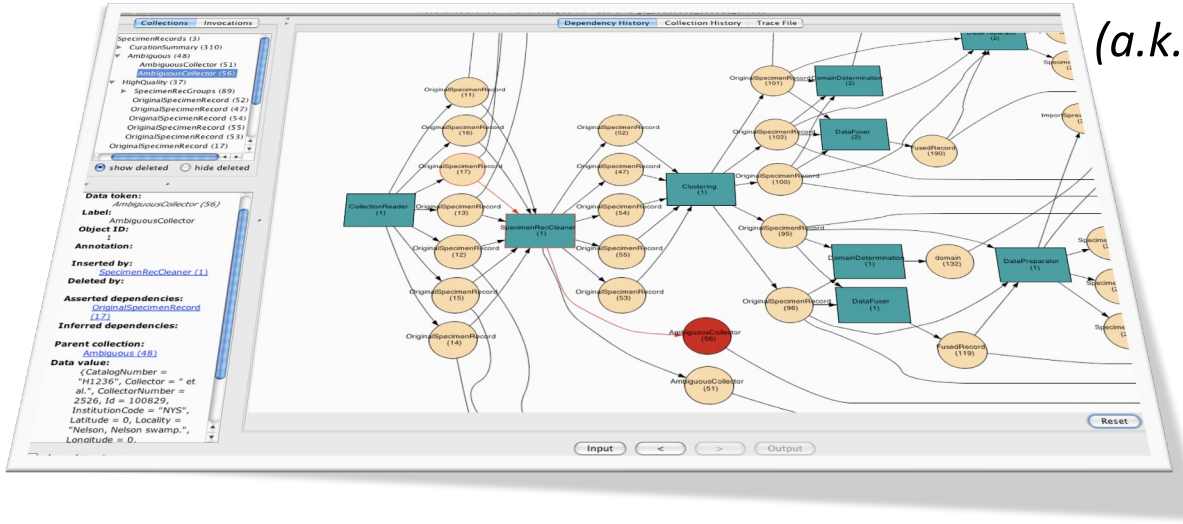
Runtime Provenance

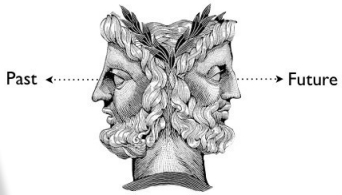
(a.k.a. ***traces, logs,***

retrospective

provenance,

"Trace-land")





Provenance Standards vs Tools

- Do we need more standards to sort this out?
  - … or do we already have too many “standards”?
- How should we **think** about provenance?
  - … in workflows and databases?
- What can we **do** with provenance?
  - … in workflows and databases?
- Tools to create, share, **use** provenance
  - … not just for “provenance for others”
  - … need more “**provenance for self**”

A simple (simplistic?) World View: ***Workflow-Land*** ⇔ ***Trace-Land***

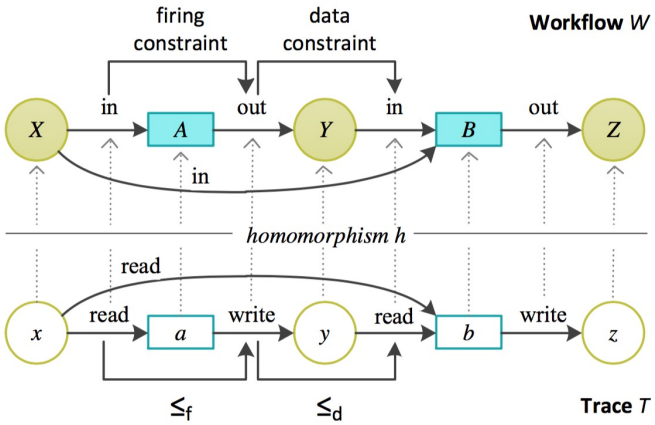
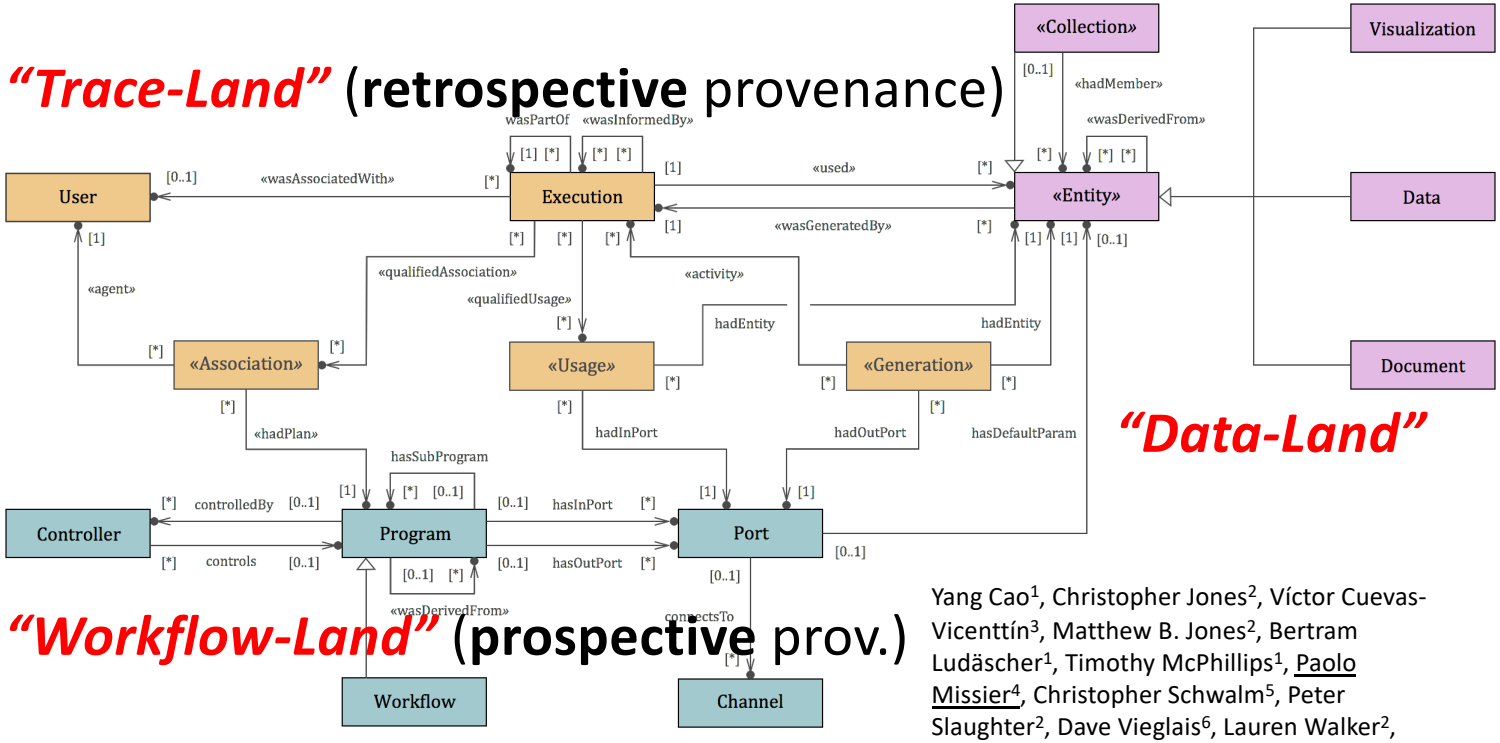


Figure 6: A homomorphism h from trace T to workflow W guarantees structural validity. Workflow-level constraints induce temporal constraints $\leq_f$ and $\leq_d$ on traces [DKBL12].

- Not a "standard" – but helps (at least me…) think about *workflows* (**prospective** provenance) and *traces* (**retrospective** provenance).

ProvONE: PROV for scientific workflows

(Transfer station to any of several other "standard extensions")



Yang Cao[1], Christopher Jones[2], Víctor Cuevas-Vicenttín[3], Matthew B. Jones[2], Bertram Ludäscher[1], Timothy McPhillips[1], Paolo Missier[4], Christopher Schwalm[5], Peter Slaughter[2], Dave Vieglais[6], Lauren Walker[2], Yaxing Wei[7]

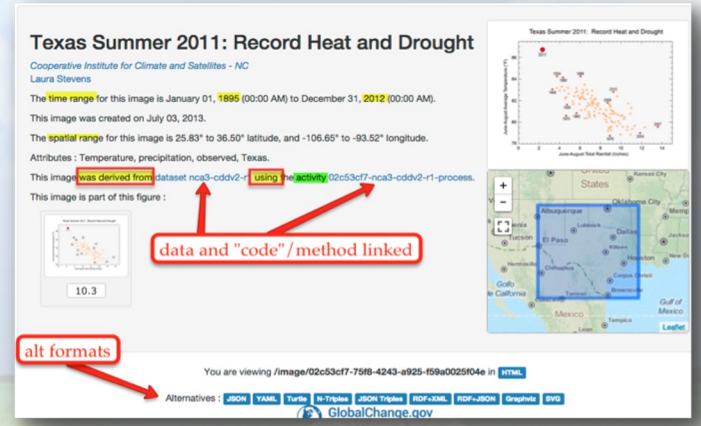
[1]University of Illinois, Urbana-Champaign, [2]National Center for Ecological Analysis and Synthesis, UCSB, [3]Universidad Popular Autónoma del Estado de Puebla, Mexico, [4]School of Computing Science,
Newcastle University, UK, [5]Woods Hole Research Center, Falmouth, MA, [6]University of Kansas, Lawrence, [7]Environmental Sciences Division, Oak Ridge National Lab, TN

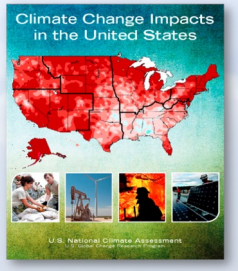
Also: A. Marinho, L. Murta, C.Werner, V.Braganholo, S. Serra da Cruz, E.Ogasawara, M. Mattoso. "ProvManager: A Provenance Management System for Scientific Workflows." Concurrency and Computation: Practice and Experience 24, no. 13 (2012): 1513–1530

…

But … how to prime the provenance pump??
Must support "**Provenance for Self**" !

Provenance for Self?!





Provenance for Others

✓ Provenance ***capture*** (Matlab, R, Python, … scientific workflow systems)

Uploading, ***sharing***, ***linking*** provenance through various provenance tools

✗ Tools for scientists to **exploit** (≠ *capture*, *share*, *link*) provenance **for their own day-to-day work.**

➔ **Prime the provenance pump** and **increase provenance generation**

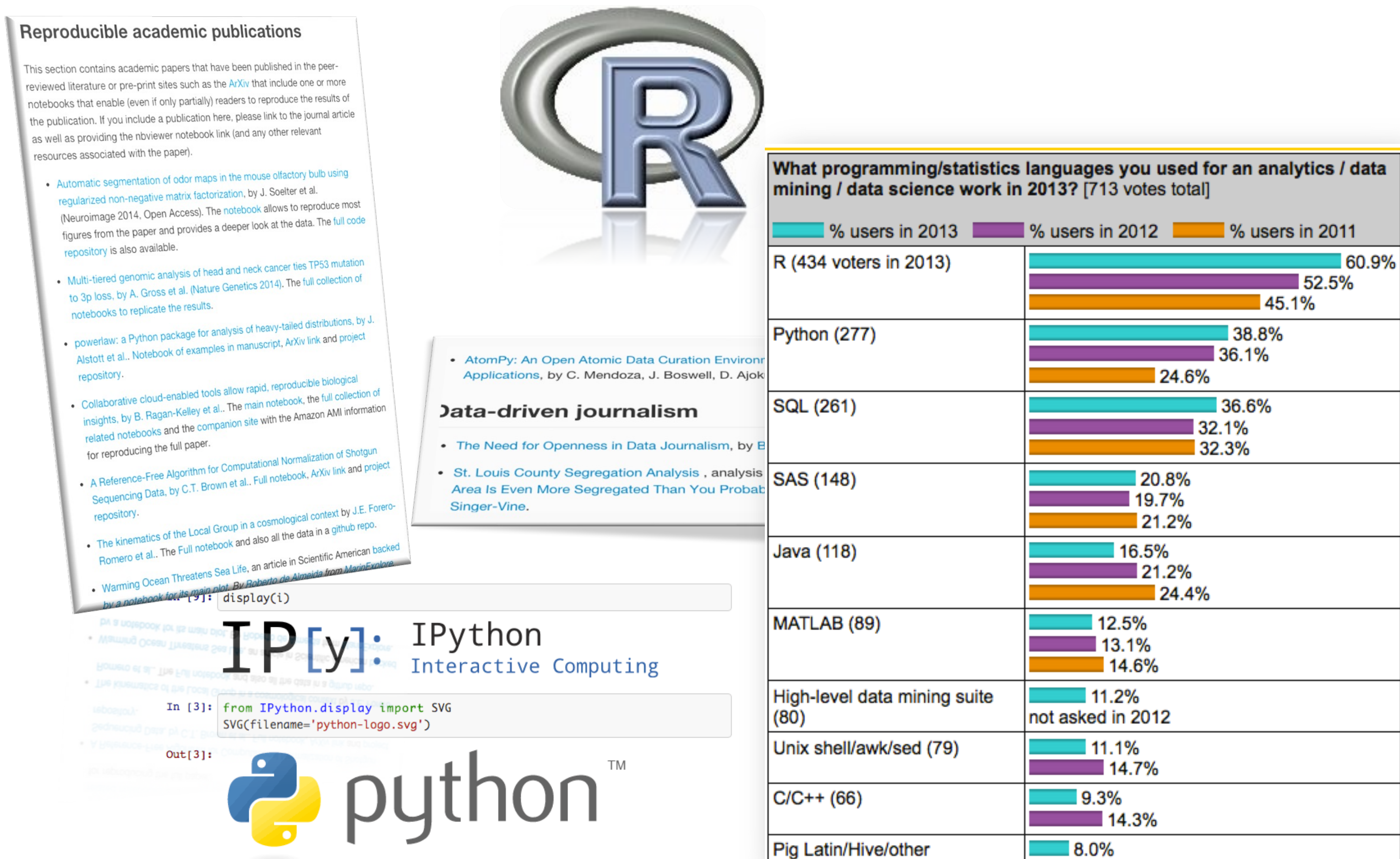
➔ Scientists **accelerate their work via new, *active* uses of provenance.**

From Workflows & Provenance to Provenance for Script-based Workflows …

- What workflow tools are (most) scientists using?
  - Workflow systems
  - … vs scripts (Python, R, MATLAB, …)

- What provenance tools are there?
  - Workflow system support
  - Tools for “workflow” scripts!?

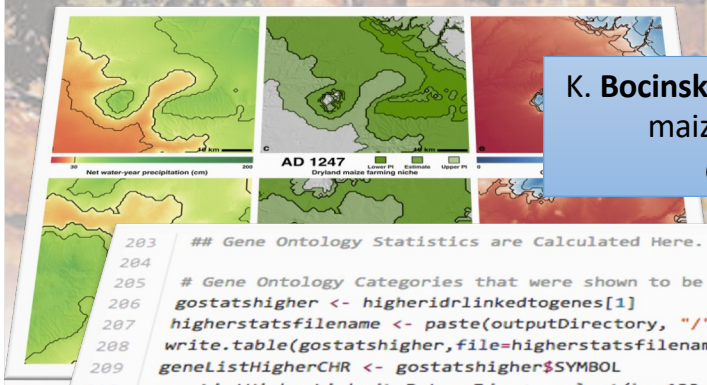
Yes, scripts are (can be) workflows too!



SKOPE: Synthesized Knowledge Of Past Environments

Bocinsky, Kohler *et al.* study rain-fed maize of **Anasazi**

- Four Corners; AD 600–1500. **Climate change** influenced **Mesa Verde Migrations**; late 13th century AD. Uses **network of tree-ring chronologies** to **reconstruct a spatio-temporal climate** field at a fairly high resolution (~800 m) from AD 1–2000. Algorithm estimates joint information in tree-rings and a climate signal to identify "best" tree-ring chronologies for climate reconstructing.



K. **Bocinsky**, T. **Kohler**, A 2000-year reconstruction of the rain-fed maize agricultural niche in the US Southwest. ***Nature Communications***. doi:10.1038/ncomms6618

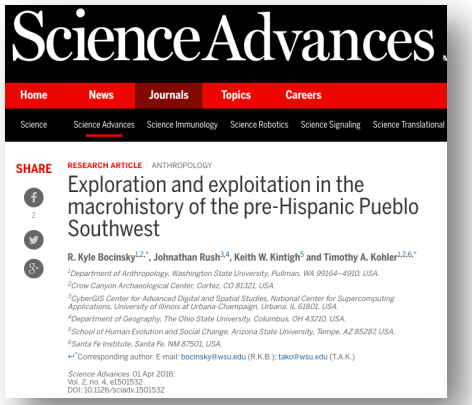
```
## Gene Ontology Statistics are Calculated Here.

 # Gene Ontology Categories that were shown to be relatively Higher (more expressed) in the Experimental Condition.
gostatshigher <- higheridrlinkedtogenes[1]
higherstatsfilename <- paste(outputDirectory, "/", runName, "_", conditions[1],"_GOStatsHigher_",mytestcond[1], "_v
write.table(gostatshigher,file=higherstatsfilename, row.names=FALSE, col.names=FALSE, quote=FALSE, sep="\t")
geneListHigherCHR <- gostatshigher$SYMBOL
geneListHigherLinkedtoEntrezIds <- select(hgu133plus2.db, keys= geneListHigherCHR, "ENTREZID", "SYMBOL")
GOstatsGenesH <- geneListHigherLinkedtoEntrezIds[,2]

x <- org.Hs.egACCNUM
mapped_genes <- mappedkeys(x)
xx <- as.list(x[mapped_genes])
geneUniverse <- (unique(names(xx)))
```

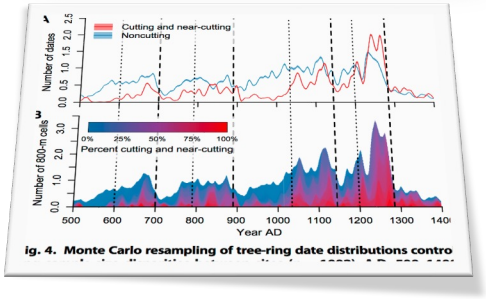
... implemented as an R Script ...

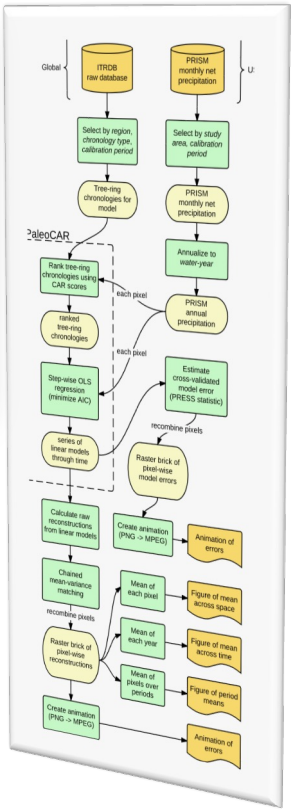
Provenance Support for Reproducible Science Example: Paleoclimate Reconstruction

Science paper (OA) uses:

- open source code:
  - R, PaleoCAR, …
- Is that all we need?
- What was the “workflow”?
- Is there prospective and/or retrospective provenance?

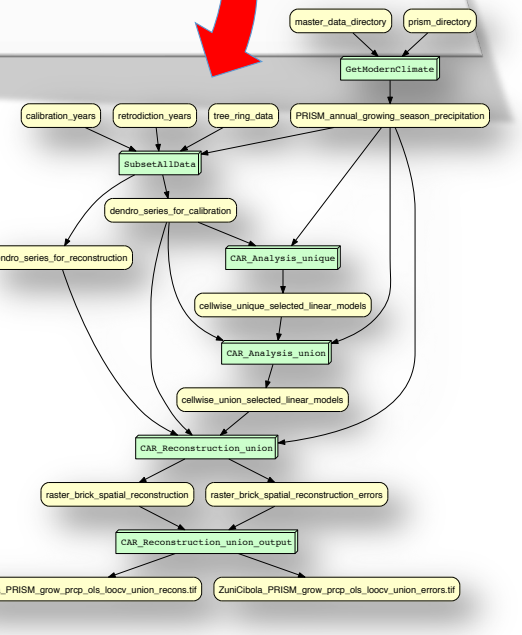






Final Tour Stop: YesWorkflow:
Yes, scripts are workflows, too!

```
203  ## Gene Ontology Statistics are Calculated Here.
204
205  # Gene Ontology Categories that were shown to be relatively Higher (more expressed) in the Experimental Condition.
206  gostatshigher <- higheridrlinkedtogenes[1]
207  higherstatsfilename <- paste(outputDirectory, "/", runName, "_", conditions[1],"_GOStatsHigher_",mytes
208  write.table(gostatshigher,file=higherstatsfilename, row.names=FALSE, col.names=FALSE, quote=FALSE, sep="\
209  geneListHigherCHR <- gostatshigher$SYMBOL
210  geneListHigherLinkedtoEntrezIds <- select(hgu133plus2.db, keys= geneListHigherCHR, "EN
211  GOstatsGenesH <- geneListHigherLinkedtoEntrezIds[,2]
212
213  x <- org.Hs.egACCNUM
214  mapped_genes <- mappedkeys(x)
215  xx <- as.list(x[mapped_genes])
216  geneUniverse <- (unique(names(xx)))
```



- **Script** *vs* Workflows/**ASAP**:
  - **A**utomation: *****
  - **S**caling: **
  - **A**bstraction: *
  - **P**rovenance: **

YW annotations: Model your Workflow!

```
1  # @BEGIN collect_data_set
2  # @PARAM cassette_id @PARAM accepted_sample @PARAM num_images @PARAM energies
3  # @OUT sample_id @OUT energy @OUT frame_number
4  # @OUT raw_image_path @AS raw_image
5  # ... @URI file:run/raw/{cassette_id}/{sample_id}/e{energy}/image_{frame_number}.raw
6  run_log.write("Collecting data set for sample {0}".format(accepted_sample))
7  sample_id = accepted_sample
8  for energy, frame_number, intensity, raw_image_path in collect_next_image(
9                          cassette_id, sample_id, num_images, energies,
10                         "run/raw/{cassette_id}/{sample_id}/e{energy}/image_{frame_number:03d}.raw"):
11     run_log.write("Collecting image {0}".format(raw_image_path))
12 # @END collect_data_set
13
14 # @BEGIN transform_images
15 # @PARAM sample_id @PARAM energy @PARAM frame_number
16 # @IN raw_image_path @AS raw_image
17 # @IN calibration_image @URI file:calibration.img
18 # @OUT corrected_image @URI file:run/data/{sample_id}/{sample_id}_{energy}eV_{frame_number}.img
19 # @OUT corrected_image_path @OUT total_intensity @OUT pixel_count
20     corrected_image_path = "run/data/{0}/{0}_{1}eV_{2:03d}.img".format(sample_id, energy, frame_number)
21     (total_intensity, pixel_count) = transform_image(raw_image_path, corrected_image_path, "calibration.img")
22     run_log.write("Wrote transformed image {0}".format(corrected_image_path))
23 # @END transform_images
```

mark the code block

… and data inputs/outputs

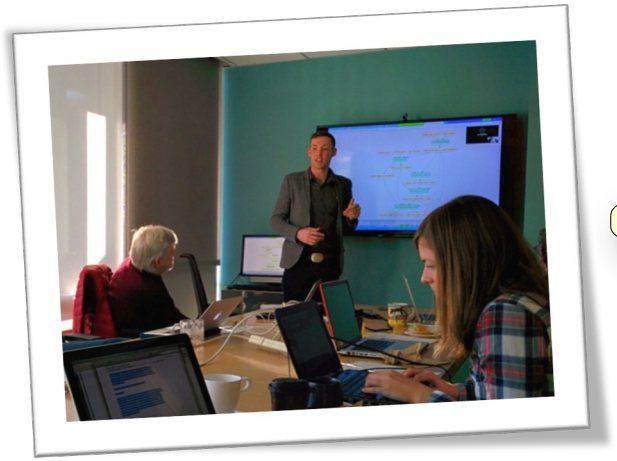
Figure 1. YW-annotated fragment of a Python script for data collection from protein crystal samples. YW-annotations `@BEGIN` and `@END` delimit code blocks; `@IN` and `@OUT` tags model relevant input and output data elements of a block; `@PARAM` identifies a block's parameters. `@URI` templates for raw images (line 5) and corrected images (line 18) link conceptual-level data elements such as `raw_image` with runtime resources (data files and their file paths). Executable script code is greyed out to emphasize YW-annotations. A program variable (`raw_image_path`) is highlighted in the code (lines 8, 11, 21): aliases (lines 4, 16) are used to link such program-level objects to the scientist's concepts (here: `raw_image`). Full example available from [MBL15].

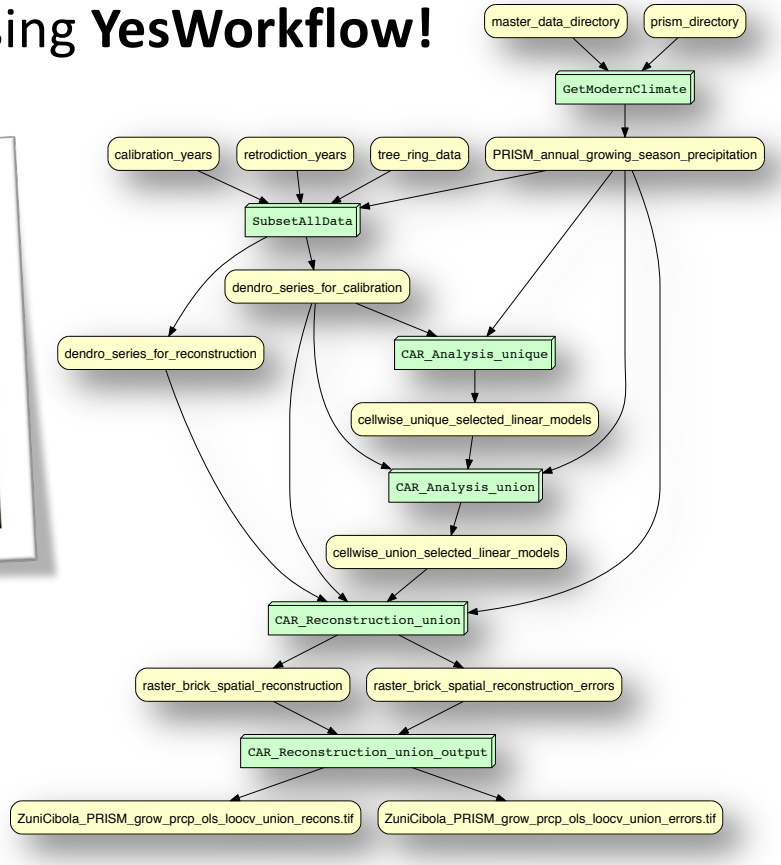
Paleoclimate Reconstruction (EnviRecon.org) ...

... explained using **YesWorkflow!**

Kyle B., (computational) archaeologist:

"It took me about 20 minutes to comment. Less than an hour to learn and YW-annotate, all-told."

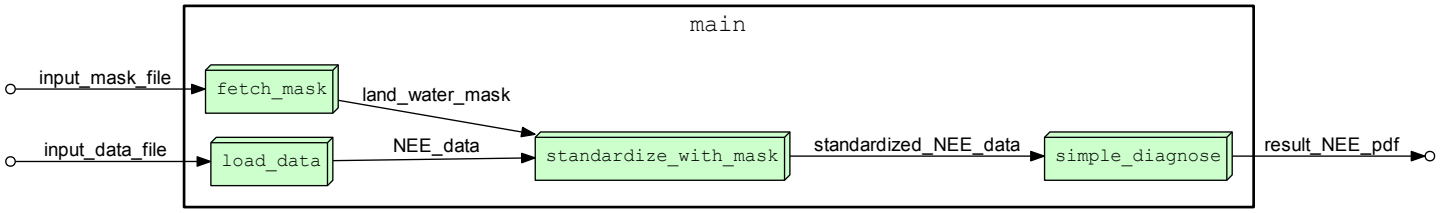




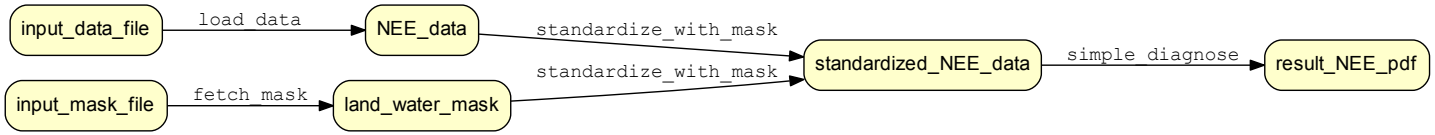
SKOPE + Kurator + DataONE (Data Observation Network for Earth)

=> YesWorkflow.org

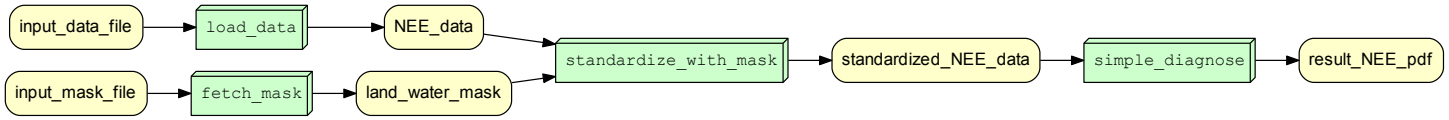
Get 3 views for the price of 1!



Process view



Data view



Combined view