CIS 602-01: Computational Reproducibility

Tools: Packaging

Dr. David Koop
Workflow Evolution Provenance of MTA Fare Data

- Initial data
  - Corrected data
    - November ff
    - November 2 data
    - August 16 Tab
  - Station locations
    - Station map
    - Added fares
    - Difference
- Sum of ffs
- 30-D weekly
- 161st-River
- Broadway line
- August 16
- Concourse line
- Heatmap
- Filtered
Workflow Evolution Provenance of MTA Fare Data

- **Initial data**
  - November ff
  - November 2 data
- **Corrected data**
  - August 16 Tab
- **Station locations**
  - Full fares map
  - Broadway line
  - Broadway diff map
  - August 16
- **Added fares**
  - Station map
  - Concourse line
  - Heatmap
  - Difference
  - Sum of ffs
  - 30-D weekly
  - 161st-River

D. Koop, CIS 602-01, Fall 2016
Workflow Evolution Provenance of MTA Fare Data

initial data

corrected data

november ff

november 2 data

August 16 Tab

station

sum of ffs

30-D weekly

161st-River

with labels

filtered

HTTPFile

CSVFile

JSONFile

JoinTables

SelectFromTable

SelectFromTable

SelectFromTable

SheetReference

SelectFromTable

ProjectTable

GMapCell

GMapCircleCell

D. Koop, CIS 602-01, Fall 2016
Workflow Evolution Provenance

[Diagram showing the workflow evolution with various nodes and connections involving data types like HTTPFile, CSVFile, JSONFile, JoinTables, SelectFromTable, ProjectTable, GMapCell, and GMapCircleCell.]
delete module “GMapCell”
delete module “CellLocation”
delete module “ProjectTable”
delete module “SelectFromTable”

...  

add module “SelectFromTable”
add parameter “float_expr” to “SelectFromTable”
with value “latitude > 40.6”
delete parameter “float_expr” from “SelectFromTable”
add parameter “float_expr” to “SelectFromTable”
with value “latitude > 40.7”
delete parameter “float_expr” from “SelectFromTable”
add parameter “float_expr” to “SelectFromTable”
with value “latitude > 40.8”

...
Execution Provenance
Execution Provenance

```xml
<module id="12" name="vtkDataSetReader"
   start_time="2010-02-19 11:01:05"
   end_time="2010-02-19 11:01:07">
   <annotation key="hash"
      value="c54bea63cb7d912a43ce"/>
</module>
<module id="13" name="vtkContourFilter"
   start_time="2010-02-19 11:01:07"
   end_time="2010-02-19 11:01:08"/>
<module id="15" name="vtkDataSetMapper"
   start_time="2010-02-19 11:01:09"
   end_time="2010-02-19 11:01:12"/>
<module id="16" name="vtkActor"
   start_time="2010-02-19 11:01:12"
   end_time="2010-02-19 11:01:13"/>
<module id="17" name="vtkCamera"
   start_time="2010-02-19 11:01:13"
   end_time="2010-02-19 11:01:14"/>
<module id="18" name="vtkRenderer"
   start_time="2010-02-19 11:01:14"
   end_time="2010-02-19 11:01:14"/>
...
```
VisTrails Demo

- Download data from web
- Process it via tools from the tabledata package
- Tag specific versions
- Run them from the command-line
Assignment 2

• [http://www.cis.umassd.edu/~dkoop/cis602/assignment2.html](http://www.cis.umassd.edu/~dkoop/cis602/assignment2.html)
• Updated this past weekend with better infrastructure to run the workflows in Docker
• Keep your project on Github, keep images on Docker Hub
• Put a link to your Docker Hub images in your Github README.md
• Note **CMD** versus **ENTRYPOINT** difference in Docker
Projects

- Survey:
  - Have not heard from many of you about the paper you will be reproducing…
  - Due tomorrow, will reply or assign

- Research:
  - How can you test if your work is reproducible?
  - Why are the specific approaches (code versioning, containers) important to your research?
  - What does your work add to existing reproducibility tools?
Covered Topics

• Version Control (Code)
• Data Sharing, Citation, and Availability
• Virtual Machines
• Containers
• Scientific Workflows
• Provenance
Tools

• We have seen specific tools, too:
  - Git, Github
  - DOIs, Dryad, DataONE, figshare
  - Xen, Parallels, EC2
  - Docker
  - Pegasus, Kepler, VisTrails, Taverna
  - PDIFF, Analogies, VisComplete

• More at ReproMatch
These are all ingredients, how do they work together?
Packaging

• If I want to replicate/reproduce something,
  - How do I ensure that I have everything I need?
  - How do I package it up so that someone else can access it?

• Two Tools
  - CDE
  - ReproZip
CDE: Using System Call Interposition to Automatically Create Portable Software Packages

P. J. Guo and D. Engler
CDE

• Mimic the specific system pieces required in a user directory
• Capture via trace
• Intercept system calls (both when capturing and when running)
• Linux-specific
ReproZip: Computational Reproducibility With Ease

F. Chirigati, R. Rampin, D. Shasha, and J. Freire
ReproZip Demo Video
ReproZip extensions to existing packaging tools

• Focus on **without-forethought** reproducibility
• Other tools: PTU, CARE, CDE
• Portability: runs using VMs or containers
• Extensibility: may port to other environments using other unpackers
• Modifiability: identifies inputs, outputs, parameters
• Usability: command-line tool and control over the trace
What should tools look like?

• "What we need right now is scientists actually using stuff that already exists, not engineers building new stuff that no one will ever use" — C. Titus Brown