Implementing Computational Reproducibility in the Whole Tale Environment

Adam Brinckman, Mary Terese Campbell, Kyle Chard, Niall Gaffney, Mihael Hategan, Matthew B. Jones, Kacper Kowalik, Sivakumar Kulasekaran, Michael Lambert, Bertram Ludäscher, Bryce Mecum, Jarek Nabrzyski, Damian Perez, Victoria Stodden, Ian Taylor, Thomas Thelen, Matthew Turk, Kandace Turner, Craig Willis and Sebastian Wyngaard

Victoria Stodden
University of Illinois at Urbana-Champaign/NCSA

2nd International Workshop on Practical Reproducible Evaluation of Computer Systems (P-RECS19)
Co-located with the 28th International Symposium on High-Performance Parallel and Distributed Computing (ACM HPDC 2019)
Phoenix, Arizona, USA - June 24th, 2019

This material is based upon work supported by the National Science Foundation under Grant No. OAC-1541450.
Whole Tale Collaboration

- **U Illinois** (NCSA) Bertram Ludäscher, Victoria Stodden, Matt Turk
  - Reproducibility; provenance / Statistics, astronomy, archaeology, crop sciences
- **U Texas, Austin** (TACC) Niall Gaffney
  - HTC; infrastructure / Bio (Cyverse) & Engineering (Design Safe)
- **U Chicago** (Globus) Kyle Chard
  - Data transfer & storage / Materials science
- **UC Santa Barbara** (NCEAS) Matt Jones
  - Publishing, provenance / Ecology & Env Sciences
- **U Notre Dame** (CRC) Jarek Nabrzyski
  - UX design; UI design / Social sciences
What is Whole Tale?

- NSF Data Infrastructure Building Block (DIBBs) project
- Building a web-based, open source platform for reproducible research for the creation, publication, and execution of tales*

- Driven by Community Engagement:
  - Working groups, internships, collaborations, etc.

- Enhancing Education & Training:
  - Training for computational reproducibility practices

*Tales are executable research objects that capture data, code, and details of the computing environment used to produce research findings.
Whole Tale: What’s in a name …

A double entendre:

- Whole tale: captures the end-to-end scientific discovery story, including computational aspects
- Whole tail: includes all computational research (e.g., long tail -- bespoke or small scale research)
Parsing Reproducibility

- **Empirical Reproducibility:**
  - traditional empirical experiments, e.g. at the bench/lab

- **Statistical Reproducibility:**
  - statistical methodology used permits generalizability of data inferences

- **Computational Reproducibility:**
  - transparency of computational steps that produce scientific findings

Simplifying **Computational Reproducibility** in Whole Tale

- Researchers can **easily package and share** tales:
  - **Data, Code, and Compute Environment**
    - .. including **narrative** and **workflow information** including inputs, outputs, and intermediates
  - to **re-create the computational results** from a scientific study
  - achieving **computational reproducibility**
  - thus “**setting the default to reproducible.**”

- Also empowers users to verify and extend results with **different** data, methods, and environments.

---

The Need for a Platform for Reproducible Research

- Enable researchers to (easily) manage the complete conduct of a computational experiment and permit its exposure as a publishable artifact
- Address the two trends simultaneously:
  - improved transparency so researchers can run much more ambitious computational experiments.
  - and better computational experiment infrastructure will allow researchers to be more transparent.

Context

- Stronger research practices
- Improved reproducibility
  - Record keeping; source code and version control; workflow management systems; tools for reproduction of results; publication reproducibility audits
- Technological and infrastructure barriers
  - Archival and open repositories; code hosting/collaboration; DOIs
- Improved replicability

https://www.nap.edu/catalog/25303/reproducibility-and-replicability-in-science
Example journal policy: American Journal for Political Science (AJPS)

- Verification policy [https://ajps.org/ajps-verification-policy/](https://ajps.org/ajps-verification-policy/)
  - "When the final draft of the manuscript is submitted, the materials will be verified to confirm that they do, in fact, reproduce the analytic results reported in the article."

- Hundreds of data/code packages already deposited to Dataverse [https://dataverse.harvard.edu/dataverse/ajps](https://dataverse.harvard.edu/dataverse/ajps)

- Third-party verification workflow (Odum Institute)

- Average review ~6 hours
Tangent:
Odum Workflow (abbreviated)

Manuscript accepted → Author submits package to Dataverse → Curator reviews package → Verifier prepares workstation

Manuscript published → Package published → Verification report sent to editor → Verifier runs/reviews outputs

Complete workflow diagram made available by Thu-Mai Christian, Assistant Director for Archives, Odum Institute for Research in Social Science
Example journal policy: ACM Transactions on Mathematical Software

- Replicated Computational Results (RCR) policy http://toms.acm.org/replicated-computational-results.cfm
  - "The authors provide the RCR reviewer access to, or sufficient description of, the computational platform used to produce the manuscript results."

- Optional additional review by referee

- RCR review report published with manuscript
Example scholarly society: American Economic Association

- Evaluating AJPS approach for AEA empirical journals
- Pre- and post-publication verification
  - Already have ~1400 published packages
  - Pre- and post-publication validation
Example university service:
CORNELL INSTITUTE for Social and Economic Research (CISER)

- Results Reproduction (R-squared)
- Service for Cornell researchers
- Computationally reproduces results of research
  - "enhanced proofreading for your Data and Code"
- Create archival package
- Average review 4-6 hours
Example voluntary/community driven "research compendia"* (in the wild)

- Materials Science https://github.com/fang-ren/Discover_MG_CoVZr/
- Climate reconstruction https://github.com/bocinsky/guedesbocinsky2018
- Computational chemistry https://github.com/sodelab/dimer-freq-manuscript
- Digital humanities https://github.com/tedunderwood/paceofchange
- ...

* I've adopted "research compendia" (Gentleman and Lang, 2004) to refer generically to the class of research objects that include Tales, Binders, Capsules, and these "homegrown" models.
Some characteristics of example research compendia

- Open access, open source
- Cite and download externally published data sources
  - Some data is private or requires authentication
- Provide multiple methods to run (local, via Docker, etc)
- Data and compendia are published to archival repositories or as supplemental information
- Use digital object identifiers for manuscript, data, and code
What exactly is (in) a Tale?

- Tale:: Research Object
  - data, code, narrative, compute environment
- Executable
- Publishable
  - Standards-based
  - Remixable
  - Transparent
  - Verifiable
Whole Tale Platform Overview

- **Authenticate** using your institutional identity
- **Access** commonly-used **computational environments**
- Easily **customize** your environment (via repo2docker)
- Reference and access externally **registered data**

- Create or upload **your data and code**
- Add **metadata** (including **provenance** information)
- Submit code, data, and environment to **archival repository**
- Get a **persistent identifier**
- **Share** for verification and re-use
Tale Creation Workflow

1. **Register externally published data by persistent identifier or upload data directly**
2. **Create a Tale, entering a name and selecting a base computational environment**
3. **A container is launched based on selected environment with an empty workspace and external data mounted read-only**
4. **Enter descriptive metadata including authors, title, description, and illustration image**
5. **Upload code, scripts, notebooks, narratives and customize the environment**
6. **Execute code/scripts to generate results/outputs**
7. **Export the Tale in compressed BagIt-RO format to run locally for verification.**
8. **Publish the Tale to an archival repository, generating a persistent identifier.**

Metadata schema:
- `schema:author`
- `schema:name`
- `schema:category`
- `pav:createdBy`
- `schema:license`
# Tale Definition and Meta Data

<table>
<thead>
<tr>
<th>Objects that comprise the Tale</th>
<th>Tale object metadata descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors: List of individual Tale authors</td>
<td>Authors: schema:author</td>
</tr>
<tr>
<td>Creators: Tale Creators (may be different than authors)</td>
<td>Creators: pav:createdBy</td>
</tr>
<tr>
<td>Title: Title of the Tale</td>
<td>Title: schema:name</td>
</tr>
<tr>
<td>Description: Description of the Tale</td>
<td>Description: schema:description</td>
</tr>
<tr>
<td>Categories: List of subject categories (keywords)</td>
<td>Categories: schema:category</td>
</tr>
<tr>
<td>Illustration: Illustration for the Whole Tale browse page</td>
<td>Identifier: schema:identifier</td>
</tr>
<tr>
<td>Create Date: Date the Tale was created</td>
<td>Illustration: schema:image</td>
</tr>
<tr>
<td>Update Date: Date the Tale was last updated</td>
<td>Create Date: pav:createdOn</td>
</tr>
<tr>
<td>License: License selected by the user</td>
<td>License: schema:license + generated file</td>
</tr>
<tr>
<td>Environment: Computational environment information</td>
<td>Environment: See environment.json e.g. *</td>
</tr>
<tr>
<td>Workspace: Code/scripts, workflow, narrative, documentation, data, results</td>
<td>Version: schema:version</td>
</tr>
<tr>
<td>External data: Data by reference to external source</td>
<td>Datasets: <a href="https://w3id.org/bundle/context">https://w3id.org/bundle/context</a></td>
</tr>
<tr>
<td>Identifier: Persistent identifier for published Tale</td>
<td>Workspace: <a href="https://w3id.org/bundle/context">https://w3id.org/bundle/context</a></td>
</tr>
</tbody>
</table>

Browse Tales

Launch to add to Launched Tales list

COMPUTATIONAL CHEMISTRY
Anharmonic vibrational structure of CO$_2$ monomer and dimer

ARCHAEOLOGY
Climate change stimulated agriculture...
Ancient farmers experienced climate changes at the local level through variations in the yields of their staple crops. However, archaeologists have had difficulty in determining where, when, and how changes in climate affected ancient farmers.

ECONOMICS
L2-Boosting for Economic Applications

Recapitulation package for: L2-Boosting for Economic Applications
The authors present the L2-Boosting algorithm and two variants, namely post-boosting and orthogonal boosting. Building

© WholeTale (Build: {commit})

This material is based upon work supported by the National Science Foundation under Grant No. OAC-1541450.
Compose New Tales
L2-Boosting for Economic Applications
Ye Luo and Martin Spindler

... Run & Interact with Tales...
### L2-Boosting for Economic Applications

**Ye Luo and Martin Spindler**

**Description**

The authors present the Local Estimation of Treatment Response (LETAR) algorithm and two variants, namely post-Boosting and orthogonal Boosting. Building on results in Ye and Spindler (2018), they demonstrate how Boosting can be used for estimation and inference of low-dimensional treatment effects. In particular, they consider estimation of a treatment effect in a setting with very many controls and in a setting with very many instruments. We provide simulations and analyze two real applications.


**Illustration**

Integration with archival repositories

- Enables turnkey exploratory data analysis on existing published datasets
- DataONE and Dataverse networks cover > 90 major research repositories!
Whole Tale Infrastructure Overview

Container orchestration (Docker Swarm)

<table>
<thead>
<tr>
<th>Core services</th>
<th>Quantitative programming environments</th>
<th>System services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy</td>
<td>Rstudio, Jupyter, Rocker</td>
<td>Monitoring</td>
</tr>
<tr>
<td>Image Registry</td>
<td></td>
<td>Logging</td>
</tr>
<tr>
<td>WholeTale Dashboard</td>
<td></td>
<td>Backup</td>
</tr>
<tr>
<td>WholeTale API</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WholeTale Filesystem</td>
<td>repo2docker wholetale</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Master Node</th>
<th>Worker 1</th>
<th>Worker 2</th>
<th>Worker 3</th>
<th>Worker n</th>
<th>Fileserver (NFS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cloud infrastructure
Key features:

Supported environments

- Extension to Binder *repo2docker*
- Jupyter, JupyterLab
- RStudio (Rocker Project)
- OpenRefine
- Forthcoming
  - Matlab, Stata
Key features:

Supported **data repositories**

- Register data from external research data repositories including Globus MDF as well as DataONE and Dataverse network members
- Referenced data is cited
  - Ideally eventually contributing to citation counts
Key features:

Publish to DataONE members

- Publish tale as DataONE package
  - KNB, Arctic Data Center
- Reference publishing implementation (v0.7)
- Issue digital object identifier
- Forthcoming
  - Dataverse network members, Zenodo
Key feature:

**Export to BagIt-RO**

- BagIt: Long-term archival format
- Re-runnable in WT
- Emerging **BagIt-RO** standard
  - Open Archival format
  - Research Object support
  - Extended for Big Data

```
tale/
  bagit.txt
  bag-info.txt
  data/
    workspace/
      run.py
      LICENSE
      requirements.txt
      output.csv
  LICENSE
  metadata/
    manifest.json
    manifest-shal.txt
  start-here/
    README.md
    tagmanifest-shal.txt
```
Key features: Export and Run Locally

- Natural outcome of Tale **export** and **repo2docker**
- Download a zip file (BagIt-RO)
- **run-local.sh**
  - Build image (**repo2docker**)
  - Fetch external data (**bdbag**)
  - Execute (**Docker**)

```
30
```
Whose problems are we addressing?

- Researchers, scientists, others ...
  - creators of tales
    - e.g. share your findings in a tale
  - reviewers of articles can review tales
    - e.g. reproduce new scientific claims
  - (re-)users of tales
    - e.g. build upon progress of others
Community Engagement

- Science & infrastructure Working Groups
- Hackathons and Workshops
- Contribute Carpentry Workshop Materials
- Provide support for RDA Early Career Fellows
- Provide yearly Internship opportunities
Looking ahead:
More **communities, data sources, publishers**

- Sharing/collaboration
- Register data from and publish to Zenodo
- Publish to Dataverse and Zenodo
- Integration with large NSF gateways:
  - CyVerse for plant and animal genomics
  - Designsafe for natural hazards civil engineering
Looking ahead:
More environments

Extend images

New base images

![Jupyter](jupyter.png) ![R Studio](r_studio.png) ![Julia](julia.png) 

![MATLAB](matlab.png) ![STATA](stata.png)
Looking ahead:
Reproducibility and Provenance

- Tracking executions
- Tracing input sources/outputs
- Exporting provenance relationships (visualization)

- Toward Really Reproducible Research
  - Transparency of computations
  - Re-execution of computations
  - Replicating and Extending computations
Looking ahead: Tales at **Scale**

Adding the ability to run tales in specialized compute environments

- Data Locality, GPU, Large memory, high throughput systems and via containerized environments
- Leveraging restful API used by existing science gateways
How is Whole Tale different?

- NSF-funded collaboration
- Focus on research repository integration
- Open source platform ("building block"): https://github.com/whole-tale
  - integrates NSF infrastructure (Jetstream, Chameleon) and partners
- Leverages Binder's repo2docker
  - Adding support for Rocker and in the future Matlab, Stata, etc.
- Working groups, internships, education/training
- Does not require Git (but allows for it)