

# CPR - A Comprehensive Provenance Record for Verification Workflows in Whole Tale

No Author Given

No Institute Given

## 1 Introduction

An increasing number of publishers of peer-reviewed journals include verification of computational artifacts in their review processes. Defining and achieving computational reproducibility has proven thorny generally, but the particular issues these publishers aim to detect are well-defined. Specific questions representative publishers aim to answer with their verification workflows include:

- Is the description in the text and supplementary materials sufficiently complete to allow others to repeat the reported computations?
- Does repeating the computations yield the reported results?

Platforms such as Binder [2] and Whole Tale [1] provide environments for assessing the reproducibility of computational artifacts by these standards via what is essentially *black-box testing* of the computational workflow: a verifier follows the instructions given in the paper to set up the required computational environment and to stage input data; triggers the start of a set of automated computations and allows them to run to completion; and finally confirms that the products of the computations match the description in the paper.

Whole Tale further aims to enable verifiers to observe *how* automated computational workflows produce intermediate and final artifacts. This will allow publishers to ask a third general question:

- Is the authors' description of the roles played by various software components comprising the overall computational workflow compatible with the actual flow of data through those components?

This capability will provide verifiers partial means for *white-box* testing of the computations reported in a paper. Specifically, a verifier will be able to detect cases where the sequence of computations and flow of data between steps does not conform to the description given in the paper.

The demonstration described in this paper exercises the software system Whole Tale is developing to enable the recording, storing, querying, and visualizing of information captured while computational workflows execute within a Tale and highlights its capabilities.

## References

1. Brinckman, A., Chard, K., Gaffney, N., Hategan, M., Jones, M.B., Kowalik, K., Kulasekaran, S., Ludäscher, B., Mecum, B.D., Nabrzyski, J., Stodden, V., Taylor, I.J., Turk, M.J., Turner, K.: Computing Environments for Reproducibility: Capturing the “Whole Tale”. *FGCS* **94**, 854–867 (2019). <https://doi.org/10.1016/j.future.2017.12.029>, <http://www.sciencedirect.com/science/article/pii/S0167739X17310695>
2. Jupyter-Project: Binder 2.0 - Reproducible, Interactive, Sharable Environments for Science at Scale. 17th Python in Science Conference (2018)