

Tools for Transparency and Replicability of Simulation in Archaeology

Mark E. Madsen

Department of Anthropology, Box 353100, University of Washington, Seattle WA, 98195 USA

Carl P. Lipo

Department of Anthropology and IIRMES, 1250 Bellflower Blvd, California State University at Long Beach, Long Beach CA, 90840 USA

Abstract

Simulation is an increasingly central tool across many theoretical frameworks but especially in evolutionary archaeology. Simulation and numerical analysis is routinely employed in hypothesis tests and model development. Simulations, however, have a well-deserved reputation as difficult to replicate and test, and it is rare that researchers beyond the authors can build upon a previously published simulation study. To improve replicability, and to make our work accessible, we employ standard tools and scripted analyses, and engage a standard software development toolchain. We describe our workflow as a contribution to best practices for simulation in archaeology.

Keywords: simulation, replicability, open methods, archaeology

1. Abstract (for ref while writing)

Simulation is an increasingly central tool across many theoretical frameworks but especially in evolutionary archaeology. Simulation and numerical analysis is routinely employed in hypothesis tests and model development. Simulations, however, have a well-deserved reputation as difficult to replicate and test, and it is rare that researchers beyond the authors can build upon a previously published simulation study. To improve replicability, and to make our work accessible, we employ standard tools and scripted analyses, and engage a standard software development toolchain. We describe our workflow as a contribution to best practices for simulation in archaeology.

Email addresses: mark@madsenlab.org (Mark E. Madsen), Carl.Lipo@csulb.edu (Carl P. Lipo)

URL: <http://madsenlab.org> (Mark E. Madsen), <http://lipolab.org> (Carl P. Lipo)

2. Notes

1. Simulations are almost never reused and built upon, and simulated data are not reused for multiple studies
 - (a) Should we expect to reuse code? Yes.
 - (b) Should we expect to reuse simulated data? Yes, under suitable circumstances.
 - (c) Example of cultural transmission models - we reimplement the same models over and over. Why?
2. Difficulties in sharing, reusing, and replicating simulations
 - (a) Baz

Reasons to reuse:

1. Iterating on model development – writing code isn't the point, building models is. At least in the exploratory stages, to the extent that we can use existing (tested) code, we can focus on the model, rather than its implementation.
2. Understanding other people's work – this doesn't have to be strict replication of results, because almost none of us spend time doing that. But I often see a model in a published paper, and want to explore its limits, or what would happen if the authors had used different combinations of parameters. In essence, I want to do model development, but starting from someone else's work.
3. Pedagogy – having students who aren't programmers reimplement most models is not productive. Except in a context where the point is to train students to write their own complex models, having students build upon existing and well documented models is probably a better way to teach both the models, and the use of simulation in contemporary science.

Reuse for any of these reasons requires that we develop and use simulation models using open methods and open tools, with an explicit eye to their reusability. Replicability is a hot issue in computational science, psychology and other social sciences (but not so much anthro), and medical research, but replicability (in the broad sense) as a goal doesn't solve everything. Transparency of operation and transparency in how the resulting data are generated by the model are crucial.

2.1. *Elements of the current approach*

1. Everything is version controlled, except **output data** which are too large to fit in a version control system.
2. Each simulation run is assigned a globally unique identifier (UUID)
3. All simulations are run in batch mode, with parameters clearly specified on each command line.
4. Each simulation run is assigned a randomly chosen seed for the random number generator, and this seed is stored WITH all output data records.

5. All output data records have key parameters stored in the record, and the simulation command used to invoke it.
6. Keep a “data dictionary” up to date with each release of the simulation software, describing the output data and metadata it creates.
7. Ensure that output data used in analysis always comes from a named, tagged release of the software, not an arbitrary checkin to source code control, so that the exact software situation can be duplicated.

2.2. Globally Unique Identifiers for Simulation Runs

Use true UUIDs (universally unique identifiers) that can be created in a decentralized fashion (i.e., without contacting a central authority). RFC1422 <http://tools.ietf.org/html/rfc4122.html> provides a specification for creating several types of UUIDs, combining accurate datetime and random elements, often with something like the hardware network address (MAC) of the machine generating the UUID. Simple libraries exist in many programming languages to generate such values. In the SeriationCT project (and others), we use the standard Python UUID library, and employ UUID-1 as the identifier variant, since it incorporates the hardware MAC address of the node executing the simulation.

The use of UUIDs as simulation identifiers virtually guarantees that a specific set of output can be tied back to the code and parameters used to create it, so long as the simulation metadata are available.

2.3. Treating Simulation Sets as Experiments

Discuss the named experiment directories, and layout of directories for simulation and analysis, storage of raw data and what happens when you mess up and need to start over.

3. Discussion

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum viverra est est. Proin eget tellus metus. Aenean ac tortor pharetra libero ultricies sagittis. Nulla facilisi. Cras tincidunt interdum tellus, quis consectetur nunc facilisis nec. Sed fermentum erat a ligula posuere quis semper risus ullamcorper. Morbi vel tincidunt augue. Nam dolor ipsum, sagittis quis dignissim eu, pulvinar sed magna. In interdum magna eu orci facilisis congue. Cras a tellus et lorem sagittis viverra. Donec risus lectus, mollis at dignissim viverra, dapibus a nulla. Vivamus porttitor scelerisque turpis, eget lobortis orci auctor eget. Donec ultricies enim ac augue porttitor convallis. Pellentesque nisl lorem, consequat a facilisis in, ornare sed lorem. In luctus, elit ac mattis dapibus, lacus elit varius tortor, vel sollicitudin massa nisl id massa. Ut sit amet nibh a sem egestas sollicitudin. Vestibulum scelerisque, dui at tincidunt accumsan, ipsum enim feugiat neque, vel interdum turpis lectus sed nisi. Nullam ultrices sodales sem, et placerat nunc euismod eu. Duis leo lacus, semper quis eleifend vitae, viverra ut nisl. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin rutrum eleifend est, id tempor velit viverra sed. Test citations: (Boyd and Richerson, 1985;

Cavalli-Sforza and Feldman, 1981; Committee on Mathematical Foundations of Verification Validation and Uncertainty Quantification, National Research Council, 2012; Ewens, 2004; Xie, 2013)

4. Acknowledgements

Lorem Ipsum

References

- Boyd, R., Richerson, P., 1985. Culture and the Evolutionary Process. University of Chicago Press, Chicago.
- Cavalli-Sforza, L., Feldman, M.W., 1981. Cultural Transmission and Evolution: A Quantitative Approach. Princeton University Press, Princeton.
- Committee on Mathematical Foundations of Verification Validation and Uncertainty Quantification, National Research Council, 2012. Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification. The National Academies Press. URL: http://www.nap.edu/openbook.php?record_id=13395.
- Ewens, W.J., 2004. Mathematical Population Genetics, Volume 1: Theoretical Introduction. 2nd ed., New York, Springer.
- Xie, Y., 2013. Dynamic Documents with R and knitr. CRC Press.