Fully automated experiments on cultural transmission through crowdsourcing

Jordan W. Suchow[1,2,3], Thomas J. H. Morgan[1,3], Jessica Hamrick[1], Michael Pacer[1], Stephan C. Meylan[1], and Thomas L. Griffiths[1,2]

The logistics of in-laboratory experiments on cultural evolution, social learning, cooperation, and collective decision-making drive experiment designs towards simple population structures, small groups, and limited interaction between participants. Here we describe Wallace, a software-based tool that automates high-throughput experiments on cultural transmission through crowdsourcing. The tool handles the full experimental pipeline from participant recruitment through data management, enabling experiments that are efficient, reproducible, and unprecedented in their complexity and scale.

The experimental study of cultural evolution, social learning, cooperation, and collective decision-making asks fundamental questions about our capacities to learn, decide, and communicate in a world that is shared with other people. Experiments have revealed, for example, how structured forms of communication emerge from individual learning and decision-making[1, 2], how innovations accumulate in populations to produce technologies that go beyond what any one individual could create[3, 4], and how the format of communication affects transmission and acquisition of new skills[5, 6]. In-laboratory experiments of this kind are logistically complex and resource intensive, requiring recruitment and coordination of participants to perform tasks sequentially and in concert, with enough space and time to isolate and control their interactions. These requirements drive experimental designs towards simple network structures, small groups, and limited interaction between participants.

To address these issues, we created a software-based tool for orchestrating cultural transmission using online crowdsourcing. Our tool, named Wallace, provides efficient high-throughput automation for running behavioral experiments — it recruits participants, obtains their informed consent, arranges them into a network, coordinates their communication, records the data they produce, pays them, and validates and manages the resulting data (Methods). Wallace runs on commodity hardware and cloud platforms, communicates by means of its API, uses widely supported languages and markups such as Python, HTML, JavaScript, and CSS, and is released as open-source software under the permissive MIT license (Methods, Supplementary

[1] Department of Psychology, University of California, Berkeley, Berkeley, California, USA. [2] Institute of Cognitive and Brain Sciences, University of California, Berkeley, Berkeley, California, USA. [3] These authors contributed equally to this work. Correspondence should be addressed to J.W.S. (suchow@berkeley.edu), T.J.H.M (thomas.j.h.morgan@gmail.com), or T.L.G. (tom_griffiths@berkeley.edu).

Software).

Wallace is modular and includes a library of components that will be useful in the creation of new experiments. Prepackaged network structures include the linear chain[9], scale-free network[10], star and burst formations, micro-society[11], and the discrete generational structure of the Wright–Fisher model from population genetics, among others. Prepackaged behavioral tasks include story recall, category learning, function learning, magnitude estimation, a public goods game, stimulus–response mapping, and numerosity judgment. Experiments can also use custom network structures, processes, and tasks, which can be built by modifying the provided templates.

To validate the extensibility of the tool, we recreated 12 experiments from evolutionary biology, game theory, and psychology, ranging from Galton's 1907 study of the wisdom of crowds[12] to a modern study of herding in humans[13] (Table 1, Methods).

Table 1. Validating Wallace's extensibility.

Topic of origin	Task	Structure, process, size	Iters.	Citation
Memory & culture	Story recall	10-person transmission chain	1	[9]
Inductive biases in learning	Function learning	10-person transmission chain	5	[14]
Wisdom of the Crowds	Magnitude estimation	100 people, unconnected		[12]
Game theory	El Farol Bar Problem	20 people, 10 rounds		
Organizational behavior	Delphi method	5-person panel with 1 overseer		
Cooperation	Iterated prisoner's dilemma			
Social learning	Replacement method	10 people, 4 active at a time		
Language	Telephone game	10-person transmission chain		[13]
Herding in humans	Numerosity judgment	10-person forward-linking chain	125	[15]
Baldwinian evolution	Category learning	60 × 40 Wright–Fisher process		
Evolution of social learning	Numerosity judgment	40 × 40 Wright–Fisher process	125	[15]
Cooperation	Public goods game	40 × 40 Wright–Fisher process	125	[15]
Design	Stimulus–response mapping	10-person transmission chain		

To validate the efficiency of Wallace, we analyzed log data from a large-scale experiment run on Wallace and reported elsewhere[15]. The experiment, which examined genetic encoding of learned behavior (i.e., the Baldwin Effect), began with a population of 60 bionic agents — human learners endowed with artificial genes that affected the success of learning. The 39 non-overlapping generations following the founding generation were each composed of a set of 60 further bionic agents, who inherited (artificial) genetic information from a member of the previous generation. Parents were chosen with probability proportional to a fitness measure that was based on their performance on the learning task (Methods). This network structure imposes a strict linear dependency across generations, but permits concurrency within each generation, such that the time complexity is linear in the number of generations and constant in the size of each generation (Supplementary Note).

Wallace makes efficient use of time, space, and human capital. Using a funnel analysis, we determined that to yield the 2400 participants called for by the design, 3300 needed to be recruited, of whom 90 (2.7%) did not begin the task, 203 (6.2%) began but quit before completion, 87 (2.6%) did not finish within the allotted time, and 460 (13.9%) finished but did not meet the required level of performance, leading to a fractional yield of 77.2% (Supplementary Note). The distribution of task completion times, which is well described as a truncated log-normal distribution (μ = 788 s, σ = 0.25 ln s, b = 1800 s), implies a minimum achievable runtime of 6.17×10^4 s (roughly 17 hours) under conditions of perfect yield and concurrent filling of a generation (Supplementary Note). The actual runtime was 2.95×10^5 s, 5.35× slower than the minimum. A blocked-time analysis[16] identified yield and upper-tail completion time as primary performance bottlenecks (Methods).

A growing concern amongst behavioral scientists is reproducibility, which is weakened by unscripted interactions with participants, small sample sizes, vaguely reported methods, unavailable source code, and undocumented data. Guided by the principle that the adoption of best practices can be promoted by the default behavior of technology, Wallace implements best practices from the ICPSR's guidelines for data preparation and archiving[17] (Methods). The tool's automation and efficiency lessen the burdens that ordinarily hinder reproducibility. For example, because the experiments are run entirely via code, they can be self-documenting, creating as a byproduct shareable packages containing the original source code, a register of all events that took place during the experiment, and the data (Methods). Wallace also offers automated preregistration (Methods).

In conclusion, Wallace is an extensible platform for automating experimentation on cultural evolution, social learning, cooperation, and collective decision-making. The tool makes efficient use of time, space, and human capital, while promoting reproducibility in the behavioral sciences. We anticipate that its greatest potential will be found in facilitating experimental designs that go beyond small linear transmission chains, leading to the proliferation and mainstreaming of paradigms such as simulating evolution with bionic agents and other forms of human-in-the-loop computation.

Methods

Methods and any associated references are available in the online version of the paper.

Acknowledgments

We thank Sally Kleinfeldt, Alec Mitchell, and Cris Ewing for discussions and assistance. This work was funded by the National Science Foundation (grants BCS-1456709 (to T.L.G and T.J.H.M.) and SPRF-IBSS-1408652 (to T.L.G and J.W.S)).

Author Contributions

All authors conceived the research. J.W.S, T.J.H.M, and J.H. wrote the software. J.W.S performed the analyses. J.W.S wrote the paper with input from all authors. All authors reviewed the manuscript.

Competing Financial Interests

The authors declare no competing financial interests.

References

1. Verhoef, T., Kirby, S. & Padden, C. in Proceedings of the 33rd annual conference of the cognitive science society 483-488 (2011).
2. Claidière, N., Smith, K., Kirby, S. & Fagot, J. Cultural evolution of systematically structured behaviour in a non-human primate. *Proceedings of the Royal Society of London B: Biological Sciences* **281**, 20141541 (2014).
3. Caldwell, C.A. & Millen, A.E. Experimental models for testing hypotheses about cumulative cultural evolution. *Evolution and Human Behavior* **29**, 165-171 (2008).
4. Dean, L.G., Kendal, R.L., Schapiro, S.J., Thierry, B. & Laland, K.N. Identification of the social and cognitive processes underlying human cumulative culture. *Science* **335**, 1114-1118 (2012).
5. Morgan, T. et al. Experimental evidence for the co-evolution of hominin tool-making teaching and language. *Nature communications* **6** (2015).
6. Hill, K.R., Wood, B.M., Baggio, J., Hurtado, A.M. & Boyd, R.T. Hunter-gatherer inter-band interaction rates: Implications for cumulative culture. (2014).
7. Flynn, E. & Whiten, A. Cultural transmission of tool use in young children: A diffusion chain study. *Social Development* **17**, 699-718 (2008).
8. Horner, V., Whiten, A., Flynn, E. & de Waal, F.B. Faithful replication of foraging techniques along cultural transmission chains by chimpanzees and children. *Proceedings of the National Academy of Sciences* **103**, 13878-13883 (2006).
9. Bartlett, F.C. Remembering: An experimental and social study. *Cambridge: Cambridge University* (1932).
10. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509-512 (1999).
11. Jacobs, R.C. & Campbell, D.T. The perpetuation of an arbitrary tradition through several generations of a laboratory microculture. *The Journal of Abnormal and Social Psychology* **62**, 649 (1961).
12. Galton, F. Vox populi (the wisdom of crowds). *Nature* **75**, 450-451 (1907).
13. Raafat, R.M., Chater, N. & Frith, C. Herding in humans. *Trends in cognitive sciences* **13**, 420-428 (2009).
14. Kalish, M.L., Griffiths, T.L. & Lewandowsky, S. Iterated learning: Intergenerational knowledge transmission reveals inductive biases. *Psychonomic Bulletin & Review* **14**, 288-294 (2007).
15. Morgan, T., Suchow, J.W. & Griffiths, T.L. Experimental evolution of human cognition through bionic simulation. (2016).
16. Ousterhout, K. et al. in Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI)(Oakland, CA 293-307 (2015).

17. Austin, E. et al. Guide to Social Science Data Preparation and Archiving Best Practice Throughout the Data Life Cycle. (2010).

Online Methods

Summary. Details of Wallace and of the reported example experiments are provided here. Additional material, including documentation for the software, is available at http://cocosci.berkeley.edu/wallace.

Code availability and licensing. The code base for Wallace version 1.0.0 is provided as Supplementary Software. It is open source and available under the MIT (X11) license, a permissive free software license. Ongoing development of Wallace and new releases of the source code are hosted on GitHub at https://github.com/berkeley-cocosci/wallace.

Command-line utility. Experiments are managed through a command-line utility, `wallace`, which includes commands to launch new experiments and monitor existing ones (Supplementary Note).

Deployment options. The software can be deployed on a local Unix-based server; on Heroku, a cloud platform-as-a-service that makes use of a managed container system (recommended); or on Amazon's EC2 cloud-computing services.

Architecture. At the center of deployed experiments is a web application built on the Flask microframework, which responds according to a versioned RESTful API that returns JSON response messages (Supplementary Note). The web application is responsible for responding to requests from the participants' front-end clients as well as to notifications from participant-handling services such as Amazon's Mechanical Turk.

File formats for storing data. During an experiment, data are stored in a PostgreSQL database, an ACID-compliant object-relational database system. Once complete, data are exported to plain-text files, one for each table in the database. Each file is formatted as comma-separated values (CSV) in accordance with a schema defined in the CSV Schema Language of The National Archives (UK), provided as Supplementary Files 1–8. These files, alongside a readme and unique identifier, are compressed into the ZIP archive file format. An example of a data archive is provided as Supplementary File 9.

File formats for storing code. The `wallace` command-line utility is launched from within a directory of code that defines the experiment to be run. At a minimum, the directory must include a configuration file in INI file format, a plain text or Markdown readme, and a Python file that defines the experiment. Examples are included within Wallace's source code. Most directories will contain additional files, including a set of HTML templates for consent forms, task instructions, etc., as well as all front-end assets (Supplementary Note).

Preregistration. To achieve preregistration, Wallace first verifies that the archived code contains a statement declaring any planned analyses. It then uses the SHA512 cryptographic hash function to compute a hexadecimal digest of the code archive at the time the experiment was run. Finally, the digest is uploaded to a publicly viewable webpage hosted by the Open Science Framework, where it is time stamped (Supplementary Note).

Objects in the Wallace universe. Wallace manages eight kinds of objects in its universe: nodes, vectors, networks, infos, transmissions, transformations, participants, and notifications. Each object is stored as an entry in a dedicated database table. A node is an agent in a particular chain or simulation. A vector is a directional connection between nodes that allows communication along it. A network is a tuple — a set of nodes and a (possibly empty) set of vectors between those nodes. An "info" is a unit of information created at a node. A transmission is an instance of information transfer along a vector. A transformation is a directional relationship between a pair

of infos that indicates when one info's contents is a function of another's (e.g., a stimulus and a response). A process is a function of the state of the Wallace universe that alters objects in it, for example by creating a connection between two nodes. A participant is a human who partakes in an experiment; each participant may have multiple associated nodes. A notification is a message sent from an outside service to Wallace — e.g., that a participant has begun the experiment.

Measures to improve yield, efficiency, and data quality. Wallace uses a combination of techniques to improve yield, efficiency, and data quality. Imperfect yield lengthens an experiment's running time because participants who do not contribute complete and valid data must be replaced. Nested failures, where a replacer needs replacing, are the performance bottleneck in cultural transmission experiments and are particularly troublesome in paradigms such as the Wright–Fisher model, where, because selection depends on relative fitness, recruitment of the next generation is contingent on having completed the parent generation. Wallace improves efficiency by screening for reliable participants, limiting the time allotted to perform the task, and testing for comprehension (Methods). Wallace achieves a speedup through a kind of apoptosis that replaced participants who use more than the time allotted to perform the task, typically set at 2–3× the predicted completion time. Unreliable participants are excluded from recruitment by requiring a minimum reputation of 95% approval on previous MTurk tasks[18].

Sourcing participants. Participants are recruited through Amazon's Mechanical Turk (MTurk), an online labor platform where people perform short tasks for pay[19-21]. With MTurk, it is possible to limit recruitment to participants from a particular geographic region. When recruitment is limited to the United States, the demographics of workers are fairly representative of the population of US internet users, though on average they are younger, have lower income, are more educated, and include more females. [Fill out description of MTurk and describe anything relevant to running experiments on it.] Experiments were approved by the Committee for Protection of Human Subjects at the University of California, Berkeley and carried out in accordance with the approved protocols.

Recruiting participants. The logic of participant recruitment is determined in part by Wallace and in part by the experimental design. The number of participants that are initially recruited is determined by the experimental design — whereas a chain starts with a single individual, the Moran process starts with an entire generation. Occasionally, a participant must be recruited to replace an existing participant. This may occur because the original participant did not begin the task, quit before completing it, did not complete it in the allotted time, finished but did not meet the required level of performance, or experienced some kind of technical error during the course of the experiment. In these cases, Wallace automatically recruits a new participant and updates its database to exclude the replaced participant from the ongoing experiment. Eventually, all the participants needed for that stage of the experiment are done, at which point a new batch of participants is recruited, as defined by the experimental design.

Compensating participants. Participants are compensated immediately following their completion of the task. Wallace allows custom logic defining the amount of compensation, making possible performance-based bonuses that depend on a participant's behavior, as is necessary for example in many experiments on cooperation.

Blocked-time analysis. A blocked-time analysis considers the counterfactual of how a system's performance would have improved if a given component had never been a bottleneck. It is similar to the engineer's calculation of efficiency, with each step in a tool chain introducing

inefficiency to the overall performance of the system. In this way, it becomes possible to identify the contribution of each component to a tool's performance. In the context of Wallace, for example, one can estimate how the running time of an experiment would have been shortened if there had been no delays in recruiting participants or if the yield had been perfect. Our analysis considered: (1) yield, (2) time to fill requests for participants, (3) length of task, (4) time allotted to complete the task, and (5) delays in sending and processing notifications.

Time efficiency. An experiment's *minimum achievable runtime* is the minimum length of time needed to complete the experiment given the dependency structure of the experiment and the time needed by a participant to complete the task.

Describe the README linter.

18. Peer, E., Vosgerau, J. & Acquisti, A. Reputation as a sufficient condition for data quality on Amazon Mechanical Turk. *Behavior Research Methods* **46**, 1023-1031 (2014).
19. Paolacci, G., Chandler, J. & Ipeirotis, P.G. Running experiments on amazon mechanical turk. *Judgment and Decision making* **5**, 411-419 (2010).
20. Buhrmester, M., Kwang, T. & Gosling, S.D. Amazon's Mechanical Turk a new source of inexpensive, yet high-quality, data? *Perspectives on psychological science* **6**, 3-5 (2011).
21. Paolacci, G. & Chandler, J. Inside the turk understanding mechanical turk as a participant pool. *Current Directions in Psychological Science* **23**, 184-188 (2014).

Supplementary information

Supplementary Note 1

Describe the time complexity of an experiment that uses a Wright–Fisher model. Describe the time vs. space tradeoffs. When running an experiment in the lab, the...

Supplementary Note 2

An experiment's *fractional yield* is the proportion of recruited participants who go on to complete the task satisfactorily. List the factors that contribute to fractional yield. Give the fractional yields from the 15 experiments described in the main paper. Plot the predicted running time of the main experiment as a function of yield. Measure yield as a function of N.

Supplementary Note 3

Describe the computation of the minimum achievable runtime.

Supplementary Note 4

Wallace can be controlled via a command-line utility, `wallace`, which has the following command available.

```
> wallace --help

Usage: wallace [OPTIONS] COMMAND [ARGS]...

  Set up Wallace as a name space.

Options:
  -h, --help  Show this message and exit.

Commands:
  create   Create a copy of the given example.
  debug    Run the experiment locally.
  deploy   Deploy app using Heroku to MTurk.
  export   Export the data.
  logs     Show the logs.
  sandbox  Deploy app using Heroku to the MTurk Sandbox.
  summary  Print a summary of a deployed app's status.
  verify   Verify that app is compatible with Wallace.
```

Further information can be found in the tutorial (Supplementary File 10).

Supplementary Note 5

Describe the directory structure of a Wallace-compatible application.

Supplementary Note 6

Give more details on preregistration.

Supplementary Note 7

Give details on time to complete and experiment runtime. Extreme value theory.

Supplementary Note 8

Supplementary Note about the reported experiment's stringent requirement w/r/t performance and what the implications for experiments with looser controls. What would the efficiency have looked like if almost everyone had passed? Do we have other experiments that we can compare with?

Figures to add:

+ Time to fill generation, which depends both on time of day and the number of participants that have already completed the task.