The Experiment Factory: Reproducible Experiment Containers

Vanessa Sochat[1]

1 Stanford University Research Computing

Summary

The Experiment Factory (Sochat 2017) is Open Source software that makes it easy to generate reproducible behavioral experiments. It offers a browsable, and tested library of experiments, games, and surveys, support for multiple kinds of databases, and robust documentation for the provided tools. A user interested in deploying a behavioral assessment can simply select a grouping of paradigms from the web interface, and build a container to serve them.

Challenges with Behavioral Research

The reproducibility crisis (Ram 2013, @Stodden2010-cu, @noauthor_2015-ig, @noauthor_undated-sn, @Baker_undated-bx, @Open_Science_Collaboration2015-hb) has been well met by many efforts (Belmann et al. 2015, @Moreews2015-dy, @Boettiger2014-cz, @Santana-Perez2015-wo, @Wandell2015-yt) across scientific disciplines to capture dependencies required for a scientific analysis. Behavioral research is especially challenging, historically due to the need to bring a study participant into the lab, and currently due to needing to develop and validate a well-tested set of paradigms. A common format for these paradigms is a web-based format that can be done on a computer with an internet connection, without one if all resources are provided locally. However, while many great tools exist for creating the web-based paradigms (Leeuw 2015, @McDonnell2012-ns), still lacking is assurance that the generated paradigms will be reproducible. Specifically, the following challenges remain:

- **Dependencies** such as software, experiment static files, and runtime variables must be captured for reproduciblity.
- Individual experiments and the library must be **version controlled.**
- Each experiment could benefit from being maintianed and tested in an **Open Source** fashion. This means that those knowledgable about the paradigm can easily collaborate on code, and others can file issues and ask questions.
- Tools must allow for **flexibility** to allow different libraries (e.g., JavaScript).
- The final product should be **easy to deploy** exactly as the creator intended.

The early version of the Experiment Factory (Sochat et al. 2016) did a good job to develop somewhat modular paradigms, and offered a small set of Python tools to generate local, static batteries from a single repository. Unfortunately, it was severely limited in its ability to scale, and provide reproducible deployments via linux containers (Merkel 2014). The experiments were required to conform to specific set of software, the lack of containerization meant that installation was challenging and error prone, and importantly, it did not meet the complete set of goals outlined above. While the `expfactory-docker` (Sochat and Blair, n.d., @Sochat2016-pu) image offered a means to deploy experiments
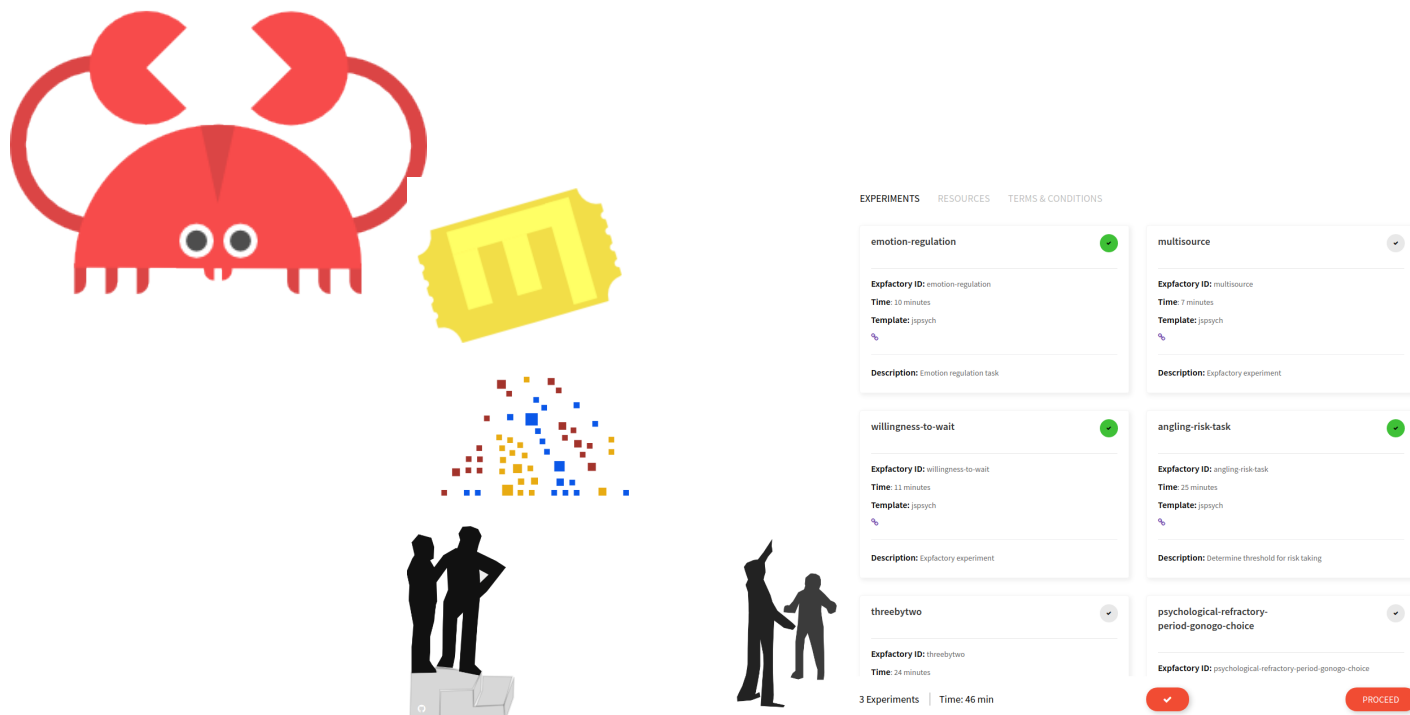
Figure 1: img/portal.png

to Amazon Mechanical Turk, it required substantial setup and was primarily developed to meet the specific needs of one lab.

Experiment Container Generation

The software outlined here, "expfactory," shares little with the original implementation beyond the name. Specifically, it allows for encapsulation of all dependencies and static files required for behavioral experimentation, and flexibility to the user for configuration of the deployment. For usage of a pre-existing experiment container, the user simply needs to run the Docker image. For generation of a new, custom container the generation workflow is typically the following:

- **Selection** The user browses a library of available experiments, surveys, and games. A preview is available directly in the browser, and data saved to the local machine for inspection. The preview reflects exactly what will be installed into the container.
- **Generation** The user selects one or more paradigms to add to the container, and clicks "Generate." The user runs the command shown in the browser on his or her local machine to produce the custom recipe for the container, called a Dockerfile.
- **Building** The user builds the container (and optionally adds the Dockerfile to version control or automated building on Docker Hub) and uses it in production. The same container is then available for others that want to reproduce the experiment.

At runtime, the user is then able to select deployment customization such as database (MySQL, PostgreSQL, sqlite3, or default of filesystem), and a study identifier.

Experiment Container Usage

Once a container is generated and it's unique identifier and image layers served in a registry like Docker Hub, it can be cited in a paper with confidence that others can run and reproduce the work simply by using it.

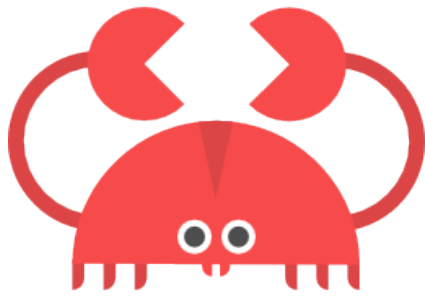More information on experiment development and contribution to the expfactory tools,

Figure 2: img/expfactory.png

containers, or library is provided at the Experiment Factory official documentation. This is an Open Source project, and so feedback and contributions are encouraged and welcome.

References

Baker, Monya. n.d. "Over Half of Psychology Studies Fail Reproducibility Test." *Nature News.*

Belmann, Peter, Johannes Dröge, Andreas Bremges, Alice C McHardy, Alexander Sczyrba, and Michael D Barton. 2015. "Bioboxes: Standardised Containers for Interchangeable Bioinformatics Software." *Gigascience* 4 (October). gigascience.biomedcentral.com:47. https://doi.org/10.1186/s13742-015-0087-0.

Boettiger, Carl. 2014. "An Introduction to Docker for Reproducible Research, with Examples from the R Environment," October. https://doi.org/10.1145/2723872.2723882.

"Docker-Based Solutions to Reproducibility in Science - Seven Bridges." 2015. https://blog.sbgenomics.com/docker-based-solutions-to-reproducibility-in-science/.

Leeuw, Joshua R de. 2015. "JsPsych: A JavaScript Library for Creating Behavioral Experiments in a Web Browser." *Behav. Res. Methods* 47 (1):1–12.

McDonnell, J V, J B Martin, D B Markant, A Coenen, A S Rich, and T M Gureckis. 2012. "PsiTurk (Version 1.02)[Software]. New York, NY: New York University."

Merkel, Dirk. 2014. "Docker: Lightweight Linux Containers for Consistent Development and Deployment." *Linux J.* Houston, TX: Belltown Media.
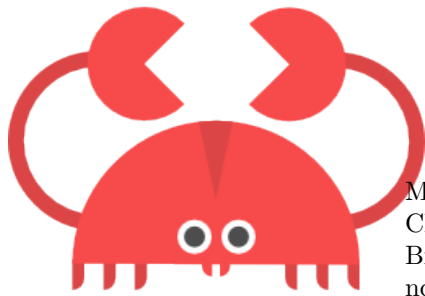
Moreews, François, Olivier Sallou, Hervé Ménager, Yvan Le Bras, Cyril Monjeaud, Christophe Blanchet, and Olivier Collin. 2015. "BioShaDock: A Community Driven Bioinformatics Shared Docker-Based Tools Registry." *F1000Res.* 4 (December). ncbi.nlm.nih.gov:1443. https://doi.org/10.12688/f1000research.7536.1.

Open Science Collaboration. 2015. "PSYCHOLOGY. Estimating the Reproducibility of Psychological Science." *Science* 349 (6251):aac4716.

Ram, Karthik. 2013. "Git Can Facilitate Greater Reproducibility and Increased Transparency in Science." *Source Code Biol. Med.* 8 (1):7.

Santana-Perez, Idafen, and María S Pérez-Hernández. 2015. "Towards Reproducibility in Scientific Workflows: An Infrastructure-Based Approach." *Sci. Program.* 2015 (February). Hindawi Publishing Corporation. https://doi.org/http://dx.doi.org/10.1155/2015/243180.

"Science Is in a Reproducibility Crisis: How Do We Resolve It?" n.d. http://phys.org/news/2013-09-science-crisis.html.

Sochat, Vanessa. 2017. "expfactory/expfactory: The Experiment Factory (v3.0) Release." https://doi.org/10.5281/zenodo.1059119.

Sochat, Vanessa V, and Ross W. Blair. n.d. "Expfactory-Docker." Github.

Sochat, Vanessa V, Ian W Eisenberg, A Zeynep Enkavi, Jamie Li, Patrick G Bissett, and Russell A Poldrack. 2016. "The Experiment Factory: Standardizing Behavioral Experiments." *Front. Psychol.* 7 (April). Frontiers.

Stodden, Victoria. 2010. "The Scientific Method in Practice: Reproducibility in the Computational Sciences," February. papers.ssrn.com.

Wandell, B A, A Rokem, L M Perry, G Schaefer, and R F Dougherty. 2015. "Data Management to Support Reproducible Research," February.