Replicability and Reproducibility in Science, and the roles played by web semantics.

Mark Gahegan
Centre for eResearch
The University of Auckland, New Zealand
m.gahegan@auckland.ac.nz

Reproducibility in science has become a very visible issue of late, with several key papers pointing out significant issues in reproducing highly cited research in the fields of psychology and bioinformatics. The movement towards *in-silico* science (simulation and modelling) means that some experiments are 'born digital' and thus should have a valid digital representation by default (though not necessarily a complete one). Other forms of science still require laboratory work (*in-vitrio*), yet even here, reproducibility is making ground.

Advanced computational workflows and scientific notebooks are now shrinking the reproducibility gap, though finally closing it represents a big challenge. Web semantics have some key roles to play in supporting truly open and reproducible science, and to this end we here (i) provide a taxonomy of approaches to reproducibility, (ii) show how semantic web technologies help to empower them, and (iii) describe some of the as yet unaddressed challenges.

Reusability and replication of analytical tasks and experiments can be supported at many levels.

Throughout this paper we adopt an approach to open, reproducible science drawn from…

- A model is **replicable** when re-running the source code produces a consistent result. *In this case, literally a digital replica of the original experiment produces the same answers.*

- A model is **reproducible** when its outputs can be reproduced by a machine from an unambiguous statement of the model equations, together with specified values of the model parameters, initial conditions and boundary conditions. In other words, the model can move from its originating source code implementation, and by means of an underlying representation based on domain theory (mathematics, logic or a mix of both) it can be successfully reproduced in some new system.

- A model is **reusable** when it can be used independently or as a module within another model. This requires that the model is well documented, the source code is available and that its limitations and appropriate use are clear. One way to describe these model features is by semantic annotation, which can provide unambiguous meaning to the variables, parameters and control sequences used. This kind of annotation requires detailed model semantics to be developed and adopted by a research community. Many research communities aspire to these kind of model semantics, which are often more difficult to capture than data semantics (and those are hard enough)

- A model is **discoverable** when it has been annotated with metadata that describe the purpose and use of the model sufficiently to allow it to be found and accessed via a webservice. Exactly what kinds of metadata are needed here is an open question. Certainly, details of the meaning of the model (see reusable) can be helpful, as can example use-cases, user feedback, statements about accuracy and some idea of the context surrounding the research task. Gahegan & Adams (2014) have some further ideas about how such semantics might be uncovered and used.

- A model is **validated** when its predictions under specified conditions match experimental observations. In other words, validation requires that we test a model against real-world observations, not just for consistency within own internal logic or mathematics. Models are typically validated within a range of 'safe' operating conditions (such as a scale interval, or between two temperature values).

https://journal.physiomeproject.org/physiome-project.html

Seeing firsthand what was done (Virtual witnessing)
At the most basic level, journals such as *JoVE* [1] can be used to capture a live account of the process of investigation, that can be peer-reviewed and shared. It does not guarantee repeatability, but it gives an insight into the mechanics of conducting research that is often missing from more traditional publications

Replicating the computational environment used
A further step towards repeatability is offered by virtualised computational infrastructure such as *Docker*[2], which offer a 'containerised' approach to supporting research. The container in this case is a place to store a computational image—a stack of software that might include an operating system, various databases, and application programs. This stack is created by serializing a working application running on a virtual machine. It has many advantages, (for example, it overcomes versioning and software integration issues for the new user) but chief amongst them for our purposes here is that the image can be moved to a completely separate virtual machine, in a different organisation or even country, where it can be opened, 're-imaged' and run in 2-3 minutes. It will behave exactly the same as the original software did, thus it provides a very convenient way to 'wrap-up' and share a complete software environment with new users. It is a mechanistic way to achieve some basic repeatability/refutability, and is mature enough now to be used reliably as part of a peer review process.

Creating a virtual library of reusable software environments
Perhaps the best example of the use of containerization for research is the *Nectar Research Cloud*[3], developed and used in Australia for the last 4-5 years (and also used by the Centre for eResearch in Auckland). As well as making it easy for researchers to create and share experiments, it also contains a huge library of existing research software images that can be easily discovered and quickly restarted. For example, one can spin up a Hadoop cluster to conduct spatial data replication experiments in just 2 minutes. Nectar has greatly increased the amount of sharing amongst Australian researchers and has been shown to enable replicability and reproducibility[4] in terms of the software used (and all the complexities and dependencies that typically plague software-reuse).

Virtual Laboratories—adding data to the software used
Of course, having access to an identical software configuration does not guarantee reproducibility or replicability, though it removes a traditionally difficult burden. But to fully replicate an analysis, the same data is also needed. A virtual Laboratory extends the idea of a Research Cloud by also including the data and macros that are used as inputs and control / conditioning elements in an analysis. The resulting environment provides a completely self-contained environment where many analytic activities become reliably repeatable, reproducible and refutable (apart from any non-deterministic methods that use randomization). An excellent example is the *Biodiversity and Climate Change Virtual Laboratory* (BCCVL)[5] which supports some very sophisticated geospatial modelling, and visualisation, but in a controlled environment that essentially wraps together all of the tools, data, methods and scripts used in analysis so they can be shared within a community. BCCVL has become a vital resource for the biodiversity research community in Australasia. Community. A similar Virtual Laboratory exists for Genomics. Virtual Laboratories need sophisticated interfaces to allow new methods and datasets to be contributed, so that they can grow to encompass new analytical methods and new data opportunities. But to do so, the methods and data need

careful curation, and in the case of methods, they must fit within specifically-designed templates in terms of how the connect together. This is a current research challenge.

Computational Workflows—flexibility for complex tasks

Where a community does not have an agreed set of methods or data, or indeed is actively developing new methods that do not easily fit into the templates used in a Virtual Laboratory, a more generic form of repeatability can be obtained using a Computational Workflow such as *Galaxy*[6]. Workflows completely describe all the analytical steps taken in an experiment or procedure, as a directed graph. They are more flexible than Virtual laboratories, in that they can create complex workflows with loops and hierarchies of analytical methods, but are also more complex to use. GeoVISTA *Studio*[7] is an early example of a workflow environment for geographical analysis and visualisation.

'Executable' Journals

Perhaps the holy grail of repeatability is a journal article that is itself an executable experiment—that describes an analysis in words, formulae and code, but also allows the analysis to be repeated by the reader. A good example is the *Physiome* journal[8] that evaluates submissions "to determine their **reproducibility**, **reusability**, and **discoverability**. At a minimum, accepted submissions are guaranteed to be in an executable state that reproduces the modelling predictions in the primary paper, and are archived for permanent access by the community." The journal uses shared method libraries, common workflow descriptions and packaged data to come good on its ambitious claims.

Caveat

All of these methods, by increasing levels of sophistication, record what was done in precise ways that can survive the process of sharing and enable researchers to reproduce the findings in a separate computational environment. However, none of them describe *why* specific choices were made by their originator, which remains an ongoing challenge.

References

[1] The Journal of Visual Experiments https://www.jove.com/

[2] https://www.docker.com/

[3] https://nectar.org.au/research-cloud/

[4] Sehrish Kanwal, Andrew Lonie, Richard O. Sinnott & Charlotte Anderson (2015). Challenges of Large-Scale Biomedical Workflows on the Cloud -- A Case Study on the Need for Reproducibility of Results. 2015 IEEE 28th International Symposium on Computer-Based Medical Systems (CBMS) (2015), Sao Carlos, Brazil, June 22, 2015 to June 25, 2015, pp: 220-225, Bookmark: http://doi.ieeecomputersociety.org/10.1109/CBMS.2015.28

[5] http://www.bccvl.org.au/

[6] https://galaxyproject.org/learn/advanced-workflow/

[7] Masa Takatsuka and Mark Gahegan (2002). GeoVISTA *Studio*: a codeless visual programming environment for geoscientific data analysis and visualization. Computers and Geosciences 28(10):1131-1144 2002.

[8] https://journal.physiomeproject.org/about.html