soundsoftware.ac.uk
Sustainable Software for Audio and Music Research

Running a Reproducible Research Prize

In this document you will find advice and considerations for running a Reproducible Research Prize (referred to as RR Prize for short). This is based on our experience in running the three rounds of RR Prizes organised by the SoundSoftware project to date.

You will find this information relevant if you intend to run a RR Prize:

- for a conference, as a member of a conference committee;
- for a conference, independently from the conference committee; or
- as an event at your institute.

The prize is primarily aimed at computational science subjects, but may also be applied to other subjects where publications' analysis and results should be reproducible.

Contents

1. Motivation

At the beginning of the SoundSoftware project, we identified four surmountable barriers to publication of research software:

- Lack of education and confidence with code
- Lack of facilities and tools
- Platform incompatibilities
- Lack of incentive, reward or recognition

The aim of running a Reproducible Research Prize is to help address the last of these barriers, kickstarting software publication by providing a clear incentive in the form of both reward and reputation.

In recognising researchers who take this extra step, or who publish work that will enable others to do so, we hope to make the idea of publication more attractive to the wider research community and to encourage people to consider publication of code when writing up their work. Ultimately, we would like this approach to spread so that it is encouraged from the early stages of a research project, both by academic supervisors and conference organisers.

2. What is a Reproducible Research Prize?

The Reproducible Research Prize is an incentive for researchers to publish the software that accompanies a paper publication, to allow other academics to reproduce the results obtained in it.

A Prize may be hosted either within an institution, or in association with a conference. Affiliation with a conference has the advantage that authors are actively engaged in preparing their papers and are well placed to adjust their work to accommodate publication and release of software as well.

We found that submissions fell into two distinct categories, for which we believe it is practical to award separate prizes:

- **Category A**: Fully reproducible work. *Awarded for a paper whose results can be reproduced using the datasets, software and methodologies described in the paper.*
- **Category B**: Reproducibility enabling work. *Awarded for a paper presenting infrastructure, datasets, or standards intended to enable future reproducible work from other authors.*

A submission for either category should consist of:

- Software (or code)
- Paper
- Dataset (where applicable)
- Instructions for installation and usage (where applicable)

Authors should be encouraged to submit their code by making it public in a generally recognised code repository and providing a link in their submission rather than by packaging their software to submit.

The submission should be reviewed and the following factors considered:

- Ease of reproducibility of the results
- Quality of sustainability planning
- Potential to enable high quality research in the research field

Note that although the aim is to incentivise broad reproducibility in general, the Prize actually only evaluates direct replicability: testing that one can obtain the same results given the same initial conditions, data, and software, rather than reproducing a whole experiment independently. The prize is thus most relevant for research topics within the Computational Science field, where software is a key component. However, it may also be applied to other subjects, for example where statistical modelling is used and would benefit from external reproducibility. The concept could also be adapted to software where reproducibility is less straightforward, for example when used in an experimental procedure obtaining real-world observations, or when high specification machines are needed. The current document describes the considerations for running a Reproducible Research Prize where basic replicability is easily tested.

3. Running a Reproducible Research Prize

3.1. Outline

The course of running the prize is split into four stages:

- Preparation - *organising the team, prize and event*
- Call for participation - *calling for submissions to the prize and arranging to receive them*
- Submission review - *reviewing the submissions*
- Presentation and Feedback - *presenting the prize and after the event*

Holding the prize in association with a peer-reviewed conference can simplify the review process, as well as increasing visibility for the prize winners.

3.2. Preparation

3.2.1. Recruiting a team

If the prize is to be hosted at a conference or society event, it might be useful to organise a voluntary chair position on the committee team, to synchronise the timetable with the other key dates of the conference and share publicity efforts with the main conference.

In any case, ideally, the lead organiser should gather a small team for the duration of the review process. As a major time commitment is only needed during the submission review stage, it may be practical to attach the organisation tasks to an existing project. Alternatively, the preparation, call for participation and presentation stages could be organised by a team of one or two, while the submission review may need additional volunteers. The review team members should have at least a basic understanding of software and system administration and, ideally, a knowledge of research within the area. Note that some resources will still be required to write up and deliver feedback to entrants after the prize has been awarded.

It is not reasonable to ask the conference paper review panel to review the software as well; reviewers will feel they have enough work already given that they are recruited on a volunteer basis. However, if the software reviewers have access to the paper reviews, they can make use of this; or the paper review panel could be asked to provide comments on some aspects needed for the Prize evaluation (see the evaluation section below).

3.2.2. What prize to offer?

The prize should be an incentive to encourage participation. Often, particularly for young academics at the beginning of the career, a certificate and the prestige may be sufficient and the “cash value” of the prize

could be nominal. This is particularly the case if the prize is associated with a conference that the authors would be submitting to anyway. We have successfully offered prizes ranging from a £100 Amazon voucher to a £1000 travel stipend to present the authors' work at an overseas conference.

An award should be offered for each separate category. Worthwhile contributions that fall slightly below the winning standard could receive an honourable mention and a certificate.

3.2.2. Preparing the Timeline

If the prize is to be hosted at a conference, the committee should be aware of the prize as early as possible, which will allow the project timeline to be synchronised with the conference key dates. That is, the initial call and submission deadline should match that of the conference. This will encourage the participants to think about reproducibility early, and so the motivation will be higher. The deadline can always be extended to shortly after that of the conference, with a repeat call, if not enough early submissions are received.

Alternatively, if the prize is not attached to a conference, previous examples have given 2 months between the call for participation and the submission deadline.

It is usually more practical to wait until the papers are accepted before reviewing the software, not least to avoid having to review papers that are then rejected from the conference. In any case, the submission review period is dependent on the number of entries. Although some entries may take minimal time to review, others may take multiple days per reviewer. This is particularly the case for submissions using large datasets or lengthy train/test cycles.

3.3. Call for participation

If the prize is being offered as part of a conference, usually the conference organisers will be able to forward a call for participation in the prize as well. Even if the prize is mentioned in the main conference call, it is worth producing a specific call for any additional information. This might include:

- Important dates
- Eligibility - *e.g. U.K. and/or non-U.K. researchers.*
- Categories
- Prizes
- Entry instructions - *may include a link to an application form, instructions or an email address to submit to.*
- Any platform limitations - *if perhaps you only have certain platforms or language environments available to test with.*
- Evaluation guide - *link to the criteria for the submission and how they are evaluated.*
- Any additional information - *e.g. submission advice.*

If you are very limited in the platforms available that you can test submissions on, you might prefer to offer something like a downloadable Linux VM image that entrants will be required to make their code compatible with, and that you can then use for testing. This should be a last resort as it may present enough of an obstacle to limit the number of submissions, especially if you are accepting submissions that use non-redistributable platform software such as MATLAB.

As it might not be clear to many what is required for high quality reproducibility, the call should include tips, hints, and possibly a minimum working example, if appropriate. A simple checklist or series of tips should help the authors detect many significant oversights, such as missing files, references to nonexistent paths (e.g. "/home/myusername/data/"), or incomplete (or completely missing) instructions.

3.4. Reviewing

The submissions should be reviewed against three separate criteria:

1. Ease of reproducibility of the results: a straightforward baseline replicability test in which you should evaluate whether each submission's published results can be regenerated using the software and data associated with the paper (where applicable).
2. Quality of sustainability planning: each submission's sustainability (of associated software and data) can be evaluated based on the Software Sustainability Institute's sustainability evaluation criteria (see http://www.software.ac.uk/online-sustainability-evaluation).
3. Potential to enable high quality research in the relevant community.

3.4.1. Ease of reproducibility

To evaluate ease of reproducibility, you will need to obtain, build, and run the software and reproduce the published results. The submission should include instructions to do this and you should approach the exercise with the expectation that little additional knowledge should be needed. If possible, use a clean test environment on an operating system that is explicitly described as supported by the submission.

Make a note of every obstacle you encounter, and consider the amount of effort and time needed to install the software. For example, if it is difficult to install any dependencies, then this should be a factor in your review. If something is obviously missing from a submission or its instructions, consider contacting the author to resolve the problem, but consider this a significant negative mark for the submission. Do not spend excessive time (say, more than one person-day) trying to make the software run at all; if you can't manage it, it's likely that other people can't either, and you may need to take up the slack if the software takes a long time to run or if other problems are encountered later.

This criterion may be evaluated using a 5-level scale as follows:

1. Excellent. A single command or script reproduced the figures in the paper.
2. Good. It was possible to generate figures like those in the paper, perhaps incomplete or with some adjustment to parameters, but without code changes or author intervention.
3. Passable. Results were generated but not without effort, for example modifying the code or reverse-engineering how to call it.
4. Modest. Although we were able to run the code, no means was provided to reproduce the figures in the paper.
5. Nil. We could not get the code to work.

3.4.2. Quality of sustainability planning

For this criterion, the Software Sustainability Institute offers a free service consisting of an online evaluation questionnaire which takes around 15 minutes to complete: http://www.software.ac.uk/online-sustainability-evaluation

Your aim is to identify those factors that may affect the ability of future users to obtain and use the software. For example, the submission should have clear licence terms, should come with sufficient documentation, should include a citation to the paper that it is associated with, and should be hosted at a recognised hosting site that can reasonably be expected to remain available.

This criterion may be evaluated on a 4-level scale:

1. Excellent. The software and/or data are clearly described in a suitable hosting environment and all of the requirements in the Software Sustainability Institute questionnaire that are relevant to the submission have been met.
2. Good. The software and/or data are sufficiently well described in a suitable hosting environment but some relevant requirements are not met.
3. Passable. The software and/or data may be obtained by future users but necessary information or supporting material is lacking.
4. Poor. It is not clear that a future user would be able to obtain or use the software or data.

3.4.3. Potential to enable high-quality research

This criterion is very useful to distinguish between submissions of similar technical merit but substantially differing relevance or ambition.

Evaluation is dependent on the qualities of your team. With a strong knowledge of the research in the chosen field, it may be possible for the review team to examine this factor themselves. Otherwise it may be more appropriate to consider recruiting volunteers such as a small review panel. If the prize is associated with a conference, then the existing peer review panel may be prepared to provide a specific score for this criterion.

A typical review scale might be 1-5 for "Very good" to "Very weak".

3.4.4. Putting the score together

There are many ways to trade off the different criteria, but it's essential to be consistent and transparent.

A workable method is:

- Reject all submissions that score worse than 3 (passable/fair) for one or more criteria
- Divide remaining submissions into the prize categories (e.g. separating fully-reproducible from reproducibility-enabling works)
- Award the prize in each category to the submission with the best (lowest) mean score across criteria

3.5. Presenting the prize

If working with a conference, it is usual to present the prize at the conference dinner or an exit talk. If you're doing this, print a certificate on nice paper so you have something to hand over, and take some publicity photos. If you're awarding "honourable mentions" for good submissions that did not quite win a prize, present them with a certificate as well.

If the winning authors are not present at the occasion then it may be necessary to post or otherwise convey the certificate and prize, but it's still important to announce the winners publicly and give them the appropriate publicity.

3.6. After the event

It's helpful to participants if you can provide them with feedback on their submissions (gently, in the case of non-winning submissions). Focus on ways that the software and code submission can be improved. Prize or no prize, the entrants want people to find, use, and cite their work, and they want to improve future work.