Stepping Stones to Reproducible Research: A Study of Current Practices in Parallel Computing[*]

Alexandra Carpen-Amarie, Antoine Rougier, and Felix D. Lübbe

Vienna University of Technology, Austria,
Faculty of Informatics, Institute of Information Systems
Research Group Parallel Computing
Favoritenstrasse 16/184-5, 1040 Vienna, Austria
{carpenamarie, rougier, luebbe}@par.tuwien.ac.at

Abstract. Experimental research plays an important role in parallel computing, as in this field scientific discovery often relies on experimental findings, which complement and validate theoretical models. However, parallel hardware and applications have become extremely complex to study, due to their diversity and rapid evolution. Furthermore, applications are designed to run on thousands of nodes, often spanning across several programming models and generating large amounts of data. In this context, reproducibility is essential to foster reliable scientific results. In this paper we aim at studying the requirements and pitfalls of each stage of experimental research, from data acquisition to data analysis, with respect to achieving reproducible results. We investigate state-of-the-art experimental practices in parallel computing by conducting a survey on the papers published in EuroMPI 2013, a major conference targeting the MPI community. Our findings show that while there is a clear concern for reproducibility in the parallel computing community, a better understanding of the criteria for achieving it is necessary.

1 Introduction

Researchers across a wide spectrum of computer science disciplines have been calling for reproducibility, as a means to assess the reliability, correctness and trustworthiness of published experimental results. This is especially the case in the area of parallel computing, where novel systems or algorithms are often backed up by computational experiments. Existing systems, tools and applications are becoming increasingly diverse and complex, typically targeting thousands of nodes, spanning across several programming models and generating large amounts of data.

A large set of papers in this domain investigate novel techniques to improve specific metrics, such as performance, and validate their approaches by performing

[*] The final publication is available at http://link.springer.com/chapter/10.1007/978-3-319-14325-5_43

experiments in customized environments. As a consequence, providing in-depth technical details regarding the implementation and experimental process to allow other researchers to reproduce their findings is of utmost importance.

The problem of experimental reproducibility is not a new one, and the scientific community has put a lot of effort into understanding the requirements of reproducible research across a variety of domains. When discussing the means to attain reproducibility in computational sciences, Peng introduces the concept of *reproducibility spectrum* to classify papers according to their degree of replicability [6]. Thus, the full replication of a study would require the availability of both source code of the proposed contribution and data originally collected by the authors to substantiate their claims. In this paper, our goal is to understand the criteria allowing us to position papers across this broad reproducibility spectrum and to examine the (often subtle) requirements for achieving reproducibility in each stage of experimental research. We target *scientific replicability*, as defined in the work of Hunold and Träff [4], that is, reproducibility of the experimental outcome, as opposed to numerical replicability, which implies bitwise reproducibility of results.

The contribution of this paper is twofold. First, we look at state-of-the-art reproducibility criteria adopted in various areas of computational research and we propose a more in-depth classification of the factors that impact the reproducibility of experimental evaluations. We argue that clearly delimitating the stages of experimental work and identifying reproducibility criteria for each of them will further facilitate both sounder research and more efficient means to review and build upon existing scientific papers. Second, we conduct a small-scale study on a series of parallel computing papers published in the 2013 EuroMPI proceedings, in order to evaluate their reproducibility potential according to our previously defined criteria.

The remainder of the paper is structured as follows. Section 2 discusses existing initiatives for reproducible research. In Section 3, we highlight the experimental stages we identified for achieving reproducible studies. We continue in Section 4 by surveying a set of published parallel computing papers to assess the weight such papers allot to each reproducibility criterion. Our experiences in attempting to replicate two of the studied papers are detailed in Section 5 and we finally point out the lessons learned in Section 6.

2 Related Work

Reproducibility is a fundamental concern in many areas of computational science, as shown by a wide range of papers addressing various facets of the subject. A set of guidelines for conducting reproducible research are stated in the work of Sandve et al. [8], who argue that reproducibility is tightly connected to the availability of all experimental details. The paper advocates for keeping track of all versions of the produced experiments, along with collected data and scripts to generate it. Vitek and Kalibera [10] investigate the root causes of non-reproducible papers in computational science, contending they can be mitigated by a careful selection of benchmarks, workloads and methodologies to document and make the experimental process available. The work of Peng and Eckel [6], despite coming from the field of biostatistics, reinforces the case for reproducibility, stating that it should be considered a "gold standard" of rigorous scientific research.

In 2012, Freire et al. [3] reviewed existing tools for computational reproducibility, proposing a three-dimensional approach for evaluating it. They argue that the level of reproducibility of an experimental paper depends on its portability to different environments, as well as on the availability of all the details related to the experimental workflow, from the source code, to the description of parameters and workload. In this paper, we focus on the same direction and we attempt to better define the criteria that characterize a fully reproducible experiment. We build on the work of Hunold and Träff [4], who first explored the state and the importance of reproducibility in parallel computing.

Several survey papers have attempted to quantify the state of reproducible research in various fields of computer science. Thus, Vandewalle et al. [9] conducted a study on a number of papers in the area of signal processing, discussing each step of their scientific methodology. With respect to the experimental workflow, the paper contends that only a small fraction of the examined papers are reproducible, although the impact of a paper seems to increase with the degree of reproducibility. Starting in 2008, the SIGMOD conference initiated an experimental reproducibility effort aiming at exploring the repeatability of experiments presented in the accepted papers [1, 5]. A large reproducibility survey in systems research has also been conducted by Collberg et al. [2], where the authors study 613 papers published in several top conferences and journals. They attempt to obtain and run the code of all experimental papers surveyed, proposing a specification system to classify papers according to the availability of their code and/or data. We take a step further and look in more detail into the methodology of reproducing experiments, as the cited studies rely on code availability as being the main factor determining reproducibility. Our study focuses on a smaller sample, that is, the proceedings of the 2013 EuroMPI conference, as a first step to investigating the reproducibility of research papers in parallel computing.

3 Criteria for Reproducible Research

This section discusses the requirements of achieving reproducible experiments in parallel computing. The scientific methodology roughly relies on four fundamental stages: (1) formulating a research question, (2) devising a hypothesis, (3) designing an experiment to test for its correctness and finally (4) analysing the results in order to draw appropriate conclusions. As experimental findings represent a huge drive of scientific discovery in parallel computing, the quality and usefulness of experimental practices is essential to understanding published contributions or extending previous research. From the standpoint of an experimenter whose goal is to verify the scientific results, providing repeatable and reproducible experiments revolves around the last two elements of the scientific method. Thus, we base our reproducibility analysis on two main steps: we will refer to experiment execution

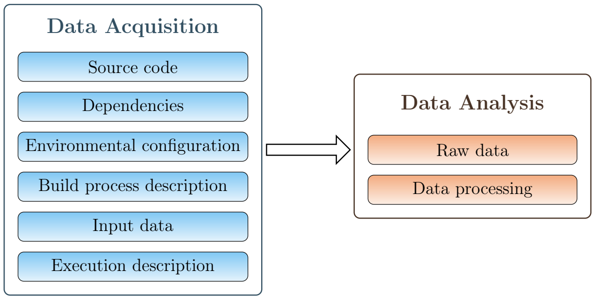

Fig. 1. Criteria for reproducible research.

as the (1) *data acquisition* step, in which all the necessary data to evaluate the studied hypothesis are generated, and to the data processing approaches that lead to the presented conclusions as the (2) *data analysis* step. The reason for partitioning reproducibility criteria into two steps lies in the fact that each of them builds on specific prerequisites, namely the availability of the source code for *data acquisition* and the the availability of the raw measured data for *analysis*.

Vandewalle et al. [9] emphasize the need for code availability, along with data and algorithm description. However, code availability is often not enough to guarantee experiment reproducibility by other researchers, as many experiments require the deployment of customized and complex software stacks. Another perspective on reproducibility criteria is provided by Peng [7], whose work focuses only on processing already measured data. Whereas he identifies several steps necessary to obtain the final results presented in scientific papers, we contend they all belong to what we call the *data analysis* step, as they depend on the availability of experimental data. Other attempts assimilate reproducibility to code availability and its successful compilation and execution [2], or focus on the same two high-level steps [1]. We propose a more nuanced look at the requirements for reproducible experiments, taking into account the impact of various fine-grained factors on reproducibility. For instance, let us consider a paper proposing a new algorithm and comparing it to state-of-the-art implementations. Even when the source code is made available in the paper, replicating the obtained results and, furthermore, extending or improving the work can be a tedious task. The readers are often forced to make their own decisions regarding compilation details, environment setup, or runtime application parameters. Such partial descriptions of the experiment may hinder the entire *data acquisition* process and result in misleading conclusions. Furthermore, the *data analysis* methodology is very often overlooked, as most papers only provide the final graphs reflecting their results and skip a detailed description allowing the reader to assess the statistical relevance of the claimed conclusions.

To investigate such issues, we propose a faceted approach to estimating the reproducibility of a given experimental workflow, by looking into several aspects for each of its stages, summarized in Figure 1.

3.1 Reproducibility Criteria for Data Acquisition

At the *data acquisition* level, we identify a set of criteria necessary for an accurate understanding and replication of an experiment:

Source code / implementation of algorithm. The availability of the source code is essential for the reproducibility of the entire experimental workflow. While papers might provide links to a source code repository, the claim for availability is not fulfilled unless the precise code version employed in the experiments is reported, and the implementation is complemented by a comprehensive documentation.

Dependent software. The majority of experimental papers rely on synthetic benchmarks or real-life applications to validate their claims, or make use of additional frameworks and libraries in order to run correctly. A key factor to verify and improve on the results is the availability of such programs. Furthermore, employed benchmarks and applications are frequently modified to highlight only specific test cases or to emphasize particular findings. Without an in-depth description of such modifications backed by the corresponding source code, other researchers and reviewers have little chance of objectively assessing the meaningfulness of experimental results.

Environmental configuration. While access to the proposed implementation is required for repeating experiments, it needs to be complemented by a rigorous description of the environment used for the study. We analyze the environment configuration by looking at three layers that can impact the experimental outcome. First, we contend that a minimal description of the employed *hardware configuration* is the key to objectively evaluate the paper conclusions. Second, the *software environment* may shape the results and provide valuable hints for a researcher attempting to reproduce the results. In particular, information concerning the operating system, the versions of installed compilers and libraries should not be overlooked. Finally, many papers disregard the need to provide *platform configuration* information along with the source code. System parameters such as process pinning, CPU frequency, or Turbo capabilities may significantly distort results and should be carefully recorded and presented.

Description of the build process. Space limitations is one of the main reasons cited for omitting code compilation details within experimental research papers. However, this is an essential step towards reproducing an experiment, and deficient descriptions may prevent researchers from being able to use the source code and associated applications. In addition, employed compilation options should be commented on, as they may have a huge impact on subsequent measurements, such as in the case of performance evaluations. Ideally, documentation should be accompanied by scripts to enable automatic parameter setup and compilation.

Description of input data. Most applications and frameworks require tuning through runtime parameters, as well as a means to initialize the processed data. Such parameters need to be documented in order to render the experiment repeatable under the same conditions. Input data sets play an equally important role as the implementation. While in some cases their investigation alone can lead to conclusions about the applicability and generality of the proposed solution, papers usually fail to thoroughly explain their workload choice.

Execution description. The manner in which the execution workflow is reported in the paper plays a key role in understanding the proposed solutions and makes it easier for other researchers to build upon them. Ideally, each scientific experiment should be complemented by an accurate depiction of the evaluation protocol. Even when the implementation of a given algorithm is available, the workflow required to test it might be complex. Thus, reproducibility calls not only for a list of employed software tools, but also for the accurate description of the means to interconnect them. Additionally, making execution scripts accessible along with the implementation facilitates the task of reproducing the experiment in a different environment.

3.2 Reproducibility Criteria for Data Analysis

Data analysis has received a lot of interest in the context of reproducible research. However, whereas in other science areas the community has imposed strict publication rules with respect to statistical significance of results, in our domain the presentation of results often lacks a detailed description of the performed underlying data processing. In this context, reproducing the data analysis phase of an experiment depends on two main criteria.

Raw data. To promote results replicability, experimental studies could make the obtained data available for other researchers interested in analysing it. Typically, it is assumed that the raw data can be generated, provided that the *data acquisition* phase is reproducible. However, this is not always the case. For instance, when the experimental settings rely on specific hardware that is not commonly accessible, a study may enable reproducibility only for the *data analysis* part by promoting publicly available data sets.

Information on data processing. A data processing step is almost always required to study raw data and to generate the final plots and tables included in the paper. It is usually an iterative process generating intermediate data sets and it consists of scripts designed to curate data, e.g., by removing outliers, to summarize, highlight particular findings and visualize them by applying statistical analysis techniques. Such scripts should be provided along with the data to substantiate the statistical significance of presented results. In their absence, reproducible *data analysis* calls for an exhaustive description of all the processing steps required to support the paper's claims. Moreover, the data processing phase often includes a qualitative analysis stage, which might be a complex process of understanding and interpreting the data, possibly requiring additional tools, such as interactive profilers or visualization tools.

4 Reproducibility Survey

To assess the state of reproducible research in parallel computing, we conducted a study on the papers published in the EuroMPI 2013 proceedings. This survey involved the 22 accepted papers, which constitute a very small and possibly not representative sample of the total number of papers published in this area. However, this survey is intended as a first attempt to study current reproducibility practices. We hope to understand the various degrees of reproducibility, as well as to better estimate the requirements of reproducible research reporting and how to improve the quality of our own research in this respect.

4.1 Overview of the Survey Process

Our survey aims at evaluating each paper against the reproducibility criteria identified in the previous section. First, for each of the 21 non-theoretical papers, we verified whether the presented experimental results are complemented with available source code. To this end, we manually scanned the papers for links to the implementation of the proposed contribution, and we additionally checked the authors' websites for further details. Furthermore, we contacted the authors to obtain access to the source code and additional information related to the other reproducibility criteria.

Next, we conducted a careful investigation for details concerning each key component of both *data acquisition* and *data analysis* phases, gathering all our findings in Table 1, which also includes the information we obtained by exchanging emails with the authors. We anonymized the collected data about the papers by listing them in random order in Table 1 We marked each reproducibility criterion with a "+", when the corresponding information was present in the paper or sent us by the authors. Papers that failed to provide any kind of information for a particular step were marked with a "○". In particular, the *Source code* and *Raw data* criteria received a "+" only if the the code/data were specifically made available in the paper. The table also comprises an estimation of the degree of reproducibility of each paper, computed as the percentage of "+" symbols that each paper has earned out of the total number of experimental stages. Moreover, the last row presents our overall results, indicating the number and the percentage of papers that include at least minimal information for each specific criterion. Some of the papers depicted experiments that did not require information related to all the previously presented criteria. In such cases, we marked the appropriate stages with a "·" and we excluded them from the computation of the overall results.

Although we anonymized the collected data, our analysis is reproducible in the sense that anyone who has access to the investigated papers is able to evaluate them with respect to the proposed reproducibility criteria. Furthermore, the data processing scripts we employed can be made available upon request.

4.2 Findings

The results in Table 1 show that a link to the source code was provided by the authors in only 30% of the papers, while information about the benchmark applications used to validate findings and input data was present in over 85% of the papers. On the other hand, the survey confirms our intuition that several

Table 1. Results of the reproducibility study on the 21 papers relying on experimental evaluations, which were published in the EuroMPI 2013 proceedings. Each reproducibility criterion is marked with a "+" if the corresponding information is included in the paper, with a "○" if the paper does not mention that particular stage, or with a "·" if such information is not needed for reproducing the paper's results.

Paper	Data acquisition						Data analysis		Repr.%
	Source code	Dep. software	Env. config.	Build process	Input data	Exec. description	Raw data	Data Processing info	
1	○	+	+	○	+	○	○	○	37%
2	+	+	·	·	+	·	○	·	75%
3	○	+	+	+	+	○	○	○	50%
4	+	+	+	+	+	+	○	+	87%
5	○	+	+	○	+	○	○	○	37%
6	+	+	+	+	+	○	○	+	75%
7	+	+	+	+	+	+	+	+	100%
8	○	+	+	○	+	+	○	+	62%
9	○	○	+	○	○	+	○	○	25%
10	○	·	+	○	○	○	○	·	16%
11	·	+	+	○	+	○	○	+	57%
12	○	+	+	○	+	+	○	○	50%
13	○	○	+	○	+	+	○	+	50%
14	○	+	+	○	+	○	○	○	37%
15	+	○	+	+	·	○	○	○	42%
16	○	+	+	+	+	○	○	○	50%
17	○	○	+	○	+	+	○	○	37%
18	○	+	+	+	+	+	○	○	62%
19	+	+	+	+	+	+	+	○	87%
20	○	○	+	○	+	○	○	○	25%
21	○	+	+	○	○	+	○	○	37%
Total	6/20 30%	15/20 75%	20/20 100%	8/20 40%	17/20 85%	10/20 50%	2/21 9%	6/19 31%	

steps of the experimental process are often neglected. Thus, only 40% of the papers mention the compilation process, while only 50% include a description of how the actual execution and measurements were performed. It is worth mentioning, however, that such low percentages are correlated to some extent with code availability. Thus, making the implementation accessible for readers would also imply devising compilation and possibly execution scripts, as well as a comprehensive documentation of these steps.

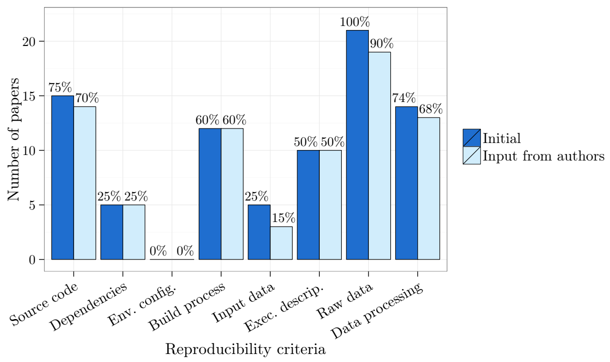

Fig. 2. Survey results: number of papers that fail to comply with each reproducibility criterion before and after contacting the authors.

Since most papers heavily rely on the experimental section, all of them provide at least a minimal description of the employed experimental environment. Each paper exhibited a hardware description of the clusters or machines used for conducting their respective studies, including informations about the architecture, processors and network interconnect. Additionally, some studies also mention the memory hierarchy and the available storage space, especially when this information is needed for further understanding and assessing the impact of the results. The software stack installed is surprisingly described in only a fraction of the considered papers, with a special focus on the employed libraries (such as MPI) and/or compilers (e.g., GCC). As the installed versions are not systematically pointed out, one could conclude that the burden of attempting to reproduce the results might be significant even for an experienced researcher. Furthermore, the information regarding specific platform configurations is even scarcer, as less than half of the papers include such details as process pinning or hyper-threading usage.

Most examined papers relied on publicly available benchmarks. Despite this fact, the high percentage corresponding to the availability of input data does not accurately account for reproducible executions, as in many cases the authors employed customized versions of the classic benchmarks and include only a brief description of the changes in the paper. Regarding the execution description, only half of the papers mention the measurement procedure or the number of repetitions and provide an insight into their workload generation approach.

Our survey unveils even more dramatic findings concerning the *data analysis* component. None of the inspected papers included a pointer to the raw data generated by their experiments, thus minimizing the possibility of further analysis by external researchers. However, in some cases we were able to obtain access to all collected data by directly contacting the authors. We contend nevertheless that the ratio of papers providing access to the raw results should be much higher. On the one hand, it would allow for a swift validation of the claimed results, provided that the data processing step is also explained. On the other hand, given the diversity and cost of today's high-performance machines, not all researchers will have access to the hardware used for the evaluation, relying only on the availability of the raw data for at least a limited reproducibility.

The data processing phase is mostly overlooked, with only 31% of the experimental papers mentioning the steps taken to analyze the data. Most papers directly provide performance figures and disregard any statistical analysis. A small subset of papers state they resort to multiple repetitions of the experiment and draw conclusions based on some method of summarizing data, such as choosing the mean or median of the obtained results.

Our findings are summarized in Figure 2, which emphasizes the number of papers that do not meet each specific criterion. The dark-colored bars show the initial results of our survey, relying only on information gathered from the published papers and the authors or projects websites, prior to contacting the authors. In contrast, the light-colored bars in Figure 2 highlight the slightly improved final values, when additional source code and data were made available for our study directly by the authors. Whereas no paper obtained a reproducibility score of 100% in our initial evaluation, the interaction with the authors led to a slightly increased value for these scores, as presented in Table 1. Nevertheless, a majority of papers do not include any information related to at least 5 of the total of 8 criteria, that is, they achieve a reproducibility score of less than 60%.

5 Case Studies: Reproducing Experiments in Practice

We selected the two papers that achieved the highest reproducibility score according to our classification and we attempted to reproduce a subset of the presented experiments. We only targeted the experiments for which we had the suitable hardware to match the setup employed by the original papers and we relied on the source code advertized within the paper as a link to the project's website or obtained by directly contacting the authors.

The findings of other reproducibility surveys [1, 2] suggest that very often compiling and running the code out-of-the-box is a time-consuming or even impossible task. In our case, we were able to obtain and compile the code, as well as to run several examples provided along with the implementation. However, code availability was not a guarantee for an effortless build process, mainly because of mismatching code versions and insufficient documentation. While this is an essential step towards reproducibility, being able to fully reproduce experiments is not limited to compiling and executing the code. The next challenge was to repeat the experiment in an environment setup similar to the one employed by the paper authors. To this end, we attempted to follow the configuration steps provided in each of the two papers.

In the first case, despite the fact that the paper details a broad range of platform and configuration data, we were unable to recreate the execution process. At this point we contacted the authors, who provided us with full access to the evaluation scripts they had been using. Thus, they supplied us with benchmarking source code, which we would have otherwise needed to re-implement based on a brief description included in the paper. We also obtained the raw data employed for the evaluation section, as well as the analysis scripts allowing us to directly generate the graphs. For the second case study, the authors provided us with both the source code and execution scripts to configure and run the experiments. Similarly to the first attempt, we also obtained the benchmarks needed for the experiments. However, conducting a similar experiment was not possible without first understanding and customizing the scripts to suit our software stack.

Interestingly, the raw data proved to be a valuable asset in our attempt to reproduce the *data acquisition* step of the experiments. While we assumed we would be able to generate our own results upon executing the experiment, having access to the original data provided by the authors helped us better understand what kind of data we needed for the next phase and what was the most appropriate format for collecting such data. Equipped with the adapted execution scripts and configuration parameters, we managed to perform the experiments on our machines effortlessly. One of the evaluated papers included the processing scripts used to generate the presented graphs. Thus, replicating the *data analysis* step was in this case equivalent to a straightforward execution of the analysis scripts on our data to obtain the corresponding figures. In the second scenario, the data processing scripts were missing, and we relied on the details reported in the paper to interpret raw data and generate figures. However, the experiments under consideration did not require complex processing steps and thus we managed to plot similar graphs to the ones depicted in the paper.

These specific examples led us to learn several important lessons. First, even though we had full access to the source code repository, we experienced compatibility problems caused by the different versions of the code we selected for our reproducibility tests. Moreover, the format of the raw data collected from our experiment did not exactly match the requirements of the processing scripts or the format of the original raw data. Consequently, we needed to manually adjust the scripts in order to generate an error-free output. It is thus essential to fully document the versions of both customized code and libraries (either within the papers or in the corresponding code/data repositories), an aspect that is often ignored in experimental descriptions. Additionally, the complexity of the software stack required for the experiments may pose problems even if all the individual details of each tool and benchmark employed are carefully presented. This was the case for one of our reproducibility tests, where the lack of execution scripts providing the correct combination of interrelated tools and benchmarks would have hindered the entire experiment replication process.

Finally, a surprising finding of our reproducing the experiments on our hardware was the fact that we obtained slightly different results from the ones depicted in the paper. We will further attempt to investigate this aspect and to identify the means to assess whether the provided results hold for the specific environment they were tested on, or they are generalizable across a wider range of platforms.

6 Conclusions

In this paper we addressed the requirements of reproducible research in terms of sound experimental practices and reporting. We discussed the impact of each of the two experimental steps, i.e., *data acquisition* and *data analysis*, on achieving reproducible results and we identified a set of reproducibility criteria matching each of them. We presented our experiences with assessing reproducibility for a selection of parallel computing papers, which helped us obtain a better understanding of the inherent difficulties of rigorous experimentation in this field. Our findings suggest that, whereas a majority of papers reflect a significant concern for reproducibility, much further work is required to attain full experimental reproducibility.

We intend to extend this survey with a wider range of papers published in this field, as well as to better understand the mechanisms for improving the state of research reproducibility at the level of each experimental stage. In our study, the compliance with a given reproducibility criterion is a binary decision. As a result, the reproducibility score we computed can only be interpreted as a measure of the minimal requirements a paper has to meet to achieve reproducibility. We plan to explore the means to fine-tune the decision for successful fulfillment of each criterion and possibly to quantify the weight of each criterion within a specific experimental scenario. Our future work will target the design of a framework for rigorously evaluating reproducibility, which can provide an objective assessment of papers in the context of reproducible research.

Acknowledgements

We would like to thank Sascha Hunold (Vienna University of Technology) for many helpful discussions and suggestions.

References

1. P. Bonnet, S. Manegold, M. Bjørling, et al. Repeatability and workability evaluation of SIGMOD 2011. *SIGMOD Record*, 40:45–48, 2011.
2. C. Collberg, T. Proebsting, et al. Measuring Reproducibility in Computer Systems Research. `http://reproducibility.cs.arizona.edu/tr.pdf`, 2014.
3. J. Freire, P. Bonnet, and D. Shasha. Computational reproducibility: State-of-the-art, challenges, and database research opportunities. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 593–596, New York, NY, USA, 2012. ACM.
4. S. Hunold and J. Larsson Träff. On the state and importance of reproducible experimental research in parallel computing. *CoRR*, abs/1308.3648, 2013.
5. I. Manolescu, L. Afanasiev, A. Arion, et al. The repeatability experiment of SIGMOD 2008. *SIGMOD Record*, 37:39–45, 2008.
6. R. D. Peng. Reproducible research in computational science. *Science*, 334(6060):1226–1227, 2011.

7. R. D. Peng and S. P. Eckel. Distributed reproducible research using cached computations. *Computing in Science and Engineering*, 11(1):28–34, 2009.
8. G. K. Sandve, A. Nekrutenko, J. Taylor, et al. Ten simple rules for reproducible computational research. *PLoS Computational Biology*, 9(10):e1003285, 2013.
9. P. Vandewalle, J. Kovacevic, and M. Vetterli. Reproducible research in signal processing. *Signal Processing Magazine, IEEE*, 26(3):37–47, 2009.
10. J. Vitek and T. Kalibera. R3: Repeatability, reproducibility and rigor. *SIGPLAN Notices*, 47(4a):30–36, 2012.