Journal of Biomedical Informatics 64 (2016) 232–254

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Identifying impact of software dependencies on replicability of biomedical workflows

Tomasz Miksa [a,*], Andreas Rauber [a], Eleni Mina [b]

[a] SBA Research & Vienna University of Technology, Wien, Austria
[b] Leiden University Medical Center, Leiden, The Netherlands

ARTICLE INFO

Article history:
Received 31 May 2016
Revised 1 September 2016
Accepted 23 October 2016
Available online 24 October 2016

Keywords:
Reproducibility
Verification
Validation
Workflow
Context model

ABSTRACT

Complex data driven experiments form the basis of biomedical research. Recent findings warn that the context in which the software is run, that is the infrastructure and the third party dependencies, can have a crucial impact on the final results delivered by a computational experiment. This implies that in order to replicate the same result, not only the same data must be used, but also it must be run on an equivalent software stack.

In this paper we present the VFramework that enables assessing replicability of workflows. It identifies whether any differences in software dependencies among two executions of the same workflow exist and whether they have impact on the produced results. We also conduct a case study in which we investigate the impact of software dependencies on replicability of Taverna workflows used in biomedical research of Huntington's disease. We re-execute analysed workflows in environments differing in operating system distribution and configuration.

The results show that the VFramework can be used to identify the impact of software dependencies on the replicability of biomedical workflows. Furthermore, we observe that despite the fact that the workflows are executed in a controlled environment, they still depend on specific tools installed in the environment. The context model used by the VFramework improves the deficiencies of provenance traces and documents also such tools. Based on our findings we define guidelines for workflow owners that enable them to improve replicability of their workflows.

1. Introduction

Research performed in Bioinformatics requires special tooling, software and processes that allow researchers to link, transform, visualise and interpret the data [1,2]. Data produced in such experiments is often reused to build new experiments [3]. However, the low maturity of tools and the possible lack of scientific scrupulousness [4] led to a low reproducibility and replicability of experiments in natural sciences in general [5,6]. Many problems can be attributed to the fact that the software is not available any more [5]. This may appear to be a pure management issue that can be overcome by imposing better policies, but recent findings show that also the context in which the software is run, that is the infrastructure and the third party dependencies, can have a crucial impact on the final results delivered by a computational experiment. In [7] the authors demonstrate that a different version of the operating system used for neuroimaging analysis in clinical research produces different visualisations. This implies that in order to replicate the same result, not only the same data must be used, but also it must be run on an equivalent software stack.

Workflow engines were proposed to bring some standardisation, as well as to hide complexity of the underlying infrastructure. Workflow engines like VisTrails [8], Kepler [9] or Taverna [10] have become popular in research areas like Astronomy or Bioinformatics [11]. They enable researchers to graphically represent their experiments in form of workflows that can be built using pre-defined elements. These elements range from dedicated data parsers to interfaces for calling external web services. Workflows can also be shared with other researchers, so that they can replicate the original experiment or reuse it. Independent peers can verify the research by re-executing the workflow and establish trust that the data produced in the experiment is correct. This in turn accelerates research, because peers have higher confidence to reuse others workflows and data.

In spite of such a standardisation, a recent study [12] reports that only 30% of almost 1500 Taverna workflows published on myExperiment [13] can be re-executed. This does not imply that

---

[*] Corresponding author.
E-mail address: miksa@ifs.tuwien.ac.at (T. Miksa).

http://dx.doi.org/10.1016/j.jbi.2016.10.011
1532-0464/

the execution produces correct results, but simply that the workflow executes. myExperiment does not provide any means to assess replicability of deposited workflows. So far there is no practice to provide data that would enable verification and validation of workflows re-executions. The provenance traces do not contain complete data describing workflow execution and there is no information on the environment in which the workflow was executed-provenance traces model relations between data and document values obtained during particular execution, but do not specify software and hardware used to process this data.

The problem thus remains on how to identify and store such data. Workflows share common infrastructure with other software running in the operating system and can delegate tasks specified in the workflow to be executed by tools installed in the environment. Such tools may require a specific configuration and presence of further tools that depend on specific software libraries or dedicated hardware. All these dependencies constitute a workflow execution context that needs to be captured, and verified and validated to state whether the workflow was replicable.

In this paper we investigate the impact of software dependencies on the replicability of biomedical workflows authored in Taverna. For that purpose we use the VFramework [14] that can verify and validate workflow re-executions. Thus, we can identify whether any differences in software dependencies among two executions of the same workflow exist and whether they have impact on the produced results. The VFramework uses the context model [15] to document environments in which the workflow executes and thus enables comparison of workflow executions without necessity of accessing both environments at the same time. By comparing context models of workflow executions we verify whether the workflow re-execution was obtained in a compliant way. The context model integrates ontologies that describe workflow and its environment. It includes not only high level description of workflow steps and services but also low level technical details on infrastructure, including hardware, software, and data. Furthermore, we use a set of automatically generated validation requirements and format-based metrics to check whether workflow re-executions produce the same results.

We conduct a case study in which we investigated the impact of software dependencies on replicability of Taverna workflows used in biomedical research for investigating the molecular mechanisms that are involved in Huntington's disease (HD). The selected workflows require multiple local dependencies ranging from additional libraries, scripts, and specific packages to external services for completing workflow steps. We investigate to what extent the impact of external services can be minimised by using service mock-ups. We also test in what way automatic workflow execution capturing helps in identifying reasons for workflows to break. We test the impact of software dependencies by re-executing the workflows not only in exact environments, but also in similar environments, that is, differing in operating system distribution and configuration. The workflows consist of steps that are representative for the majority of Taverna workflows published on myExperiment [12]. Despite the fact, that our discussion is focused on Taverna workflows, the challenges and ways of addressing them remain valid for other workflow systems as well, differing in the actual technical implementation.

The paper is structured as follows. In Section 2 we present related work on workflow management systems, provide an overview of reproducibility and replicability challenges, and discuss the context of scientific experiment including means of its capturing and storing. Section 3 provides an overview of the VFramework. Section 4 describes steps of the VFramework that capture the original workflow execution, while Section 5 describes the steps that verify and validate the workflow re-execution. In Section 6 we present a case study from the biomedical research domain in which we applied the VFramework to identify the impact of software dependencies on the reproducibility of workflows. In Section 7 we provide the discussion of the results and provide recommendations. Conclusions are in Section 8.

2. Related work

In the following section we describe examples of related work. We first discuss different workflow management systems and then we provide an overview of studies related to replicability of computational studies that emphasise the necessity to document the workflow execution context and show that no sufficient verification and validation method is in use. Finally, we review in what way the software execution context can be modelled and automatically captured.

2.1. Workflow management systems

Workflow engines like VisTrails [8], Kepler [9] or Taverna [10] have become popular in research areas like astronomy or bioinformatics. They enable researchers to graphically represent their experiments in form of workflows that can be built using predefined elements. These elements range from dedicated data parsers to interfaces for calling external web services. A comparison of workflow management systems is presented in [16]. The authors compared Taverna, Kepler and Activiti [17]. The first two are typically used in scientific settings, while Activiti is used in business settings, because it allows modelling of workflows using the Business Process Modelling Notation (BPMN). The results of the comparison are summarised in Table 1. The compared systems were implemented in Java and each of them supports scripting languages. Taverna allows Beanshells that are based on Java, Kepler allows for Python scripts, while Activiti supports a wide range of different scripting languages. Each of the workflow systems records the workflow provenance in a database. The traces contain information specifying who ran the experiment at what point in time, what data was provided as input and what intermediate data was exchanged between the workflow steps.

Due to the similarities among workflow engines, we chose one workflow system on which we applied the VFramework for verification and validation of workflow re-executions. For this reason the discussion presented in this dissertation is focused on Taverna workflows, but the challenges and ways of addressing them remain valid for other workflow systems as well, differing only in the actual technical implementation. We also applied the VFramework to experiments in other technical settings. These workflows did not require workflow engines because they were modelled as bash scripts or python scripts, still working successfully. However, these examples are beyond the scope of this paper.

2.2. Reproducibility and replicability challenges

The RDA working group on dynamic data citation[1] produced recommendations [18] on how to deal with changing datasets. One of the recommendations says that the data must be timestamped and all insert and delete actions must be marked so that the state of the database at the moment when the experiment was conducted could be restored. Otherwise running the same workflow with different input data means running a different experiment. The replicability [19] of research assumes, that we can obtain the same results in the same conditions, in our case that we can obtain the same result using the same inputs and identical/compatible environment. Reproducibility can be understood as obtaining the similar results in

---

[1] https://rd-alliance.org/groups/data-citation-wg.html.