FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

A Provenance-based Semantic Approach to Support Understandability, Reproducibility, and Reuse of Scientific Experiments

Dissertation
zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)

vorgelegt dem Rat der Fakultät für Mathematik und Informatik
der Friedrich-Schiller-Universität Jena

von Sheeba Samuel
geboren am 10.10.1989 in Adoor, Indien

Gutachter

1. **Prof. Dr. Birgitta König-Ries**
   Friedrich-Schiller-Universität Jena, 07743 Jena, Deutschland

2. **Prof. Dr. Carole Goble**
   The University of Manchester, M13 9PL Manchester, UK

3. **Prof. Dr. Harald Sack**
   FIZ Karlsruhe - Leibniz Institute for Information Infrastructure, 76344 Eggenstein-Leopoldshafen, Deutschland

Tag der öffentlichen Verteidigung: 20. Dezember 2019

Ehrenwörtliche Erklärung

Hiermit erkläre ich,

- dass mir die Promotionsordnung der Fakultät bekannt ist,
- dass ich die Dissertation selbst angefertigt habe, keine Textabschnitte oder Ergebnisse eines Dritten oder eigenen Prüfungsarbeiten ohne Kennzeichnung übernommen und alle von mir benutzten Hilfsmittel, persönliche Mitteilungen und Quellen in meiner Arbeit angegeben habe,
- dass ich die Hilfe eines Promotionsberaters nicht in Anspruch genommen habe und dass Dritte weder unmittelbar noch mittelbar geldwerte Leistungen von mir für Arbeiten erhalten haben, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen,
- dass ich die Dissertation noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prfung eingereicht habe.

Bei der Auswahl und Auswertung des Materials sowie bei der Herstellung des Manuskripts haben mich folgende Personen unterstützt:

- Prof. Dr. Birgitta König-Ries
- Prof. Dr. H. Martin Bücker

Ich habe die gleiche, eine in wesentlichen Teilen ähnliche bzw. eine andere Abhandlung bereits bei einer anderen Hochschule als Dissertation eingereicht: Ja / [✓ Nein].

Jena, den 20. Dezember 2019

[Sheeba Samuel]

To
Samuel and Leelamma, my parents for making me who I am today
John, my brother for being my role model
Thomas, my husband for everything

Acknowledgements

I would like to express my deepest gratitude to my advisor Prof. Dr. Birgitta König-Ries. She created an environment with all the guidance and support to accomplish this work. She not only believed in me but also allowed me to approach things in my own way. I would like to thank her for her encouraging words and giving unprecedented opportunities to expand my research horizon. I would like to thank my co-supervisor Prof. Dr. Martin Bücker for providing helpful suggestions, especially during writing. I also owe special thanks to Prof. Dr. Carole Goble and Prof. Dr. Harald Sack, who served as external reviewers for my dissertation and provided very valuable feedback.

I would like to express my sincere gratitude to all the ReceptorLight project members. Interactions with them helped me in formulating and implementing many components presented in this thesis. I owe special thanks to Prof. Dr. Christoph Biskup and Dr. Kathrin Groenevald for helping me shape this research work. I am also thankful to Tom Kache, Teresa Langenstück and Bastian Bunzeck for their support in my evaluation studies. I would also like to thank all the participants of the evaluation study conducted as part of this thesis.

I appreciate the support that I received from my colleagues Daniel Walther and Frank Taubert. I would like to express my sincere gratitude to my incredible colleagues in the Fusion group. My thanks go not only for their support in my research work but also for making work more fun. I also thank all my colleagues who supported me by providing feedback for my research and presentations.

Thanks to DFG for providing financial support through the ReceptorLight project. I am also thankful to Friedrich-Schiller University Jena for providing me ProChance Grant to support my conference travels.

I would like to express my sincere gratitude to Jodi Schneider, Michelle Cheatham, and Vojtech Svatek for providing me valuable feedback through ESWC PhD Symposium paper reviews. I would also thank all the people I met during conferences and the fruitful discussions I had with them.

My special thanks go to all my past teachers and professors since my childhood for their encouragement. A warm-hearted thanks to Anthony, Alice, Priya, Shahina, Sanchita, Ruchi, and Sahad for their support, love, and care.

Most importantly, I am indebted to my family. I thank my parents who continuously

supported my decisions and encouraged me to always look forward. I would like to express my heartfelt gratitude to my brother, Dr. John Samuel. He's always there for me with personal and professional guidance since my childhood. Special thanks to him for reading my thesis and providing valuable suggestions. I would also thank my in-laws for their love and care. Lastly, I would like to thank my loving husband, Thomas, for being there for me and making my life merrier.

Abstract

Understandability and reproducibility of scientific results are vital in every field of science. The scientific community is interested in the results of experiments which are understandable, reproducible and reusable. Recently, there is a rapidly growing awareness in different scientific disciplines on the importance of reproducibility. Several reproducibility measures are being taken to make the data used in the publications findable and accessible. However, these measures are usually taken when the papers are published online. But, there are many challenges faced by scientists from the beginning of an experiment to the end in particular for data management. The explosive growth of heterogeneous research data and understanding how this data has been derived is one of the research problems faced in this context. Provenance, which describes the origin of data, plays a key role to tackle this problem by helping scientists to understand how the results are derived. Interlinking the data, the steps and the results from the computational and non-computational processes of a scientific experiment is important for the reproducibility. The lack of tools which address this requirement fully is the driving force behind this research work.

Working towards this goal, we introduce the notion of "end-to-end provenance management" of scientific experiments to help scientists understand and reproduce the experimental results. The main contributions of this thesis are: (1) We propose a provenance model "REPRODUCE-ME" to describe the scientific experiments using semantic web technologies by extending existing standards. (2) We study computational reproducibility and important aspects required to achieve it. (3) Taking into account the REPRODUCE-ME provenance model and the study on computational reproducibility, we introduce our tool, ProvBook, which is designed and developed to demonstrate computational reproducibility. It provides features to capture and store provenance of Jupyter notebooks and helps scientists to compare and track their results of different executions. (4) We provide a framework, CAESAR (**C**oll**A**borative **E**nvironment for **S**cientific **A**nalysis with **R**eproducibility) for the end-to-end provenance management. This collaborative framework allows scientists to capture, manage, query and visualize the complete path of a scientific experiment consisting of computational and non-computational steps in an interoperable way. We apply our contributions to a set of scientific experiments in microscopy research projects.

Zusammenfassung

Verständlichkeit und Reproduzierbarkeit der wissenschaftlichen Ergebnisse sind in jedem Bereich der Wissenschaft unerlässlich. Die wissenschaftliche Gemeinschaft ist vorrangig an den Ergebnissen von Experimenten interessiert, die verständlich, reproduzierbar und wiederverwendbar sind. Aktuell ist eine deutliche Steigerung des Bewusstseins für die Bedeutung der Reproduzierbarkeit in verschiedenen wissenschaftlichen Disziplinen zu beobachten. Es werden verschiedene Maßnahmen zur Reproduzierbarkeit ergriffen, um die in den Publikationen verwendeten Daten zum Zeitpunkt ihrer Online-Veröffentlichung auffindbar und zugänglich zu machen. Dabei stehen Forschende jedoch im gesamten Verlauf des Experiments vor vielen Herausforderungen, insbesondere bezüglich des Datenmanagements. Der Umgang mit dem explosionsartigen Wachstum heterogener Forschungsdaten und die Gewährleistung der Nachvollziehbarkeit der Datenherkunft sind Forschungsthemen, mit denen Wissenschaftler in diesem Kontext konfrontiert sind. Provenance beschreibt den Ursprung der Daten. Sie spielt eine Schlüsselrolle bei der Bewältigung dieser Problems, da sie den Beteiligten hilft zu verstehen, wie Ergebnisse abgeleitet werden. Die Notwendigkeit, die Daten, die Verarbeitungsschritte und die Ergebnisse der computergestützen sowie der nicht-computergestützten Prozesse eines wissenschaftlichen Experiments miteinander zu verknüpfen, ist für die Reproduzierbarkeit wichtig. Der Mangel an Werkzeugen, die bei der vollständigen Erfüllung dieser Anforderung unterstützen, ist die Motivation zu dieser Forschungsarbeit.

Um dieses Ziel zu erreichen, führen wir den Begriff des "End-to-End Provenance Managements" wissenschaftlicher Experimente ein, der die Tätigkeit beschreibt, die sicherstellt, dass Forschende dass experimentelle Ergebnisse veständlich und reproduzierbar sind. Die wichtigsten Beiträge dieser Arbeit sind: (1) Wir schlagen das Provenance-Modell "REPRODUCE-ME" vor, um wissenschaftliche Experimente mithilfe von Semantic Web-Technologien unter Einbeziehung bestehender Standards zu beschreiben. (2) Wir untersuchen die Reproduzierbarkeit von Berechnungen und wichtiger Aspekte, die dazu erforderlich sind. (3) Wir stellen das Werkzeug ProvBook vor, das zum Nachweis der Eignung des REPRODUCE-ME Provenance-Modells und zur Bestätigung der Resultate der Studie zur rechnerischen Reproduzierbarkeit entwickelt wurde. ProvBook erfasst und speichert die Herkunft der Ausführung von Jupyter-Notebooks und hilft Wissenschaftler/innen, ihre

Ergebnisse verschiedener Ausführungen zu vergleichen und zu nachzuvollziehen. (4) Wir stellen CAESAR (**C**oll**A**borative **E**nvironment for **S**cientific **A**nalysis with **R**eproducibility), ein kollaboratives Framework für das end-to-end Provenance Management, vor. CAESAR ermöglicht, es Wissenschaftler/innen, den gesamten Weg eines wissenschaftlichen Experiments, das aus rechnerischen und nicht rechnerischen Schritten besteht, interoperabel zu erfassen, zu verwalten, abzufragen und zu visualisieren. Wir wenden unsere Beiträge auf eine Reihe wissenschaftlicher Experimente in Forschungsprojekten der Mikroskopie an.

Contents

List of Figures

Listings

List of Tables

Chapter 1

Introduction

With the advent of things like sensors, satellites, microscopes that can produce more and more data and things like computers that can process more and more data, the way that science is being done has dramatically changed. This change has happened in the real as well as in the virtual world. As a result, it has become more complex to keep track of how the experimental results are derived. This is important because scientific experiments play an increasingly important role in coming up with the new findings and in extending the knowledge of the world. The increasing magnitude of data produced in the experiments and understanding how the results are derived from them brings several old and new challenges to the light. Reproducibility is one such challenge which has always been discussed in science even in the time of Galileo (1564-1642) [Atmanspacher and Maasen, 2016]. And it is still under discussion with more concerns towards the "Reproducibility Crisis" [Kaiser, 2015,Peng, 2015,Begley and Ioannidis, 2015,Baker, 2016,Hutson, 2018] in this 21st century which is driven by computational science.

A survey conducted by Nature in 2016 among 1576 researchers brought greater insight into the reproducibility crisis [Baker, 2016]. According to the survey, 70% of researchers have tried and failed to reproduce other scientists' experiments. The main reasons for the irreproducible research as cited in Baker's paper involve selective reporting, the pressure to publish, poor analysis, unavailability of methods and code, etc. They also mention that this crisis is different across domains. As per the survey, more scientists in biology and pharmaceutical industry agreed that there is a significant reproducibility crisis than the scientists from computer science or physics. Reproducibility is one of the criteria for scientists in trusting the published results. Though it is a very complex concept[1], it does not have a common global standard definition among all fields of science. This results in having different research works and measures to enable reproducibility across disciplines.

Recently, several reproducibility measures are taken by different organizations to tackle this problem along with the new research works in this area. The National

[1] The exact definition used in this thesis can be found in Chapter 4.

Institute of Health (NIH) announced in 2016 the "Rigor and Reproducibility" guidelines[2] to support reproducibility in biomedical research. Journals like Nature make it mandatory to have the data used for experiments mentioned in the publications to be findable and accessible. In 2014, Nature introduced a condition for publication requiring the authors to "*make materials, data, code, and associated protocols promptly available to readers without undue qualifications*"[3]. The FAIR principles have been introduced in this regard to enable findability, accessibility, interoperability, and reuse of data [Wilkinson et al., 2016]. However, these measures are taken at the top level when scientific papers are published online. Figure 1.1 shows the different levels in a scientific research study. We consider the research lifecycle of scientists from the data acquisition to the publication of results as a pyramid. The bottom level consists of data acquisition where the data is collected from different sources. The terms bottom, ground or grass root level will be used interchangeably throughout this thesis. The sources can range from manual surveys and interviews to image acquisition from a microscope. The data collection phase is followed by data processing. The processed data is then curated and later analyzed for results. The final results are eventually published to the scientific community. As we go up the levels in the pyramid, the size of the data decreases. At the top level, publications share only a subset of the data which is collected at the bottom level.

It is not only important to take reproducibility measures at the top level but also at grass root level during data creation by the scientists either working individually or collaboratively. One of the main challenges faced at this level is the management and exchange of experiments along with all the data required for understandability, reproducibility, and reuse.

In order to reproduce own results or other scientist's results, it is essential to know the methods and steps taken to generate the output. A key factor to support scientific reproducibility is the "provenance" information which tells about the origin or history of the data. Recording and analysis of provenance data of a scientific experiment play a vital role for scientists to repeat or reproduce [Taylor and Kuyatt, 1994] any experiment. Preservation of data and research methods is an important requirement to track the provenance of results. At the same time, it is also required to represent and express this information in an interoperable way so that scientists can understand the data and results.

In this thesis, we aim to address how to support understandability, interoperability, and reproducibility of experimental results. In order to do so, we bring together the concepts of provenance [Herschel et al., 2017] and semantics [Berners-Lee et al., 2001]. We examine the role of semantic web technologies for the end-to-end prove-

[2] `https://www.nih.gov/research-training/rigor-reproducibility/principles-guidelines-reporting-preclinical-research`, Accessed on March 17, 2019.

[3] `https://www.nature.com/authors/policies/availability.html`, Accessed on March 17, 2019.

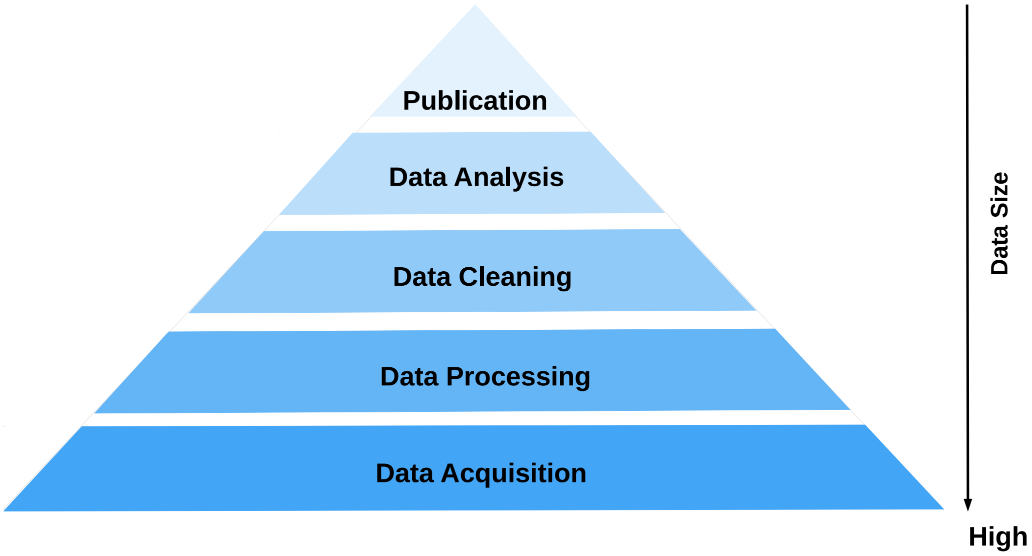


Figure 1.1: Different levels in a scientific research study

nance management of experiments to track the complete path towards their results. The ultimate goal of this research is to provide a well-structured framework for the preservation and accessibility of the experimental data and its provenance starting from the bottom level of a research study. In order to achieve this aim, we focus on the development of computational tools to help scientists capture, represent, manage and visualize the complete path taken for their experiments.
This chapter presents the motivation behind our research work in Section 1.1. We provide basic terminology of the concepts that we use throughout this thesis in Section 1.2. Section 1.3 presents an overview of our contributions. The structure of the thesis is outlined in Section 1.4 which is followed by the list of publications that have been published as parts of the works described in this thesis (Section 1.5).

1.1 Motivation

The concrete motivation for our work arises from the requirement to develop a platform for the management and modeling of experimental data for the Collaborative Research Center (CRC) ReceptorLight[4]. Scientists from two universities[5], two university hospitals[6] and a non-university research institute[7] work together to understand the function of membrane receptors and develop high-end light microscopy techniques. Membrane receptors, an important factor in all biological processes,

[4] `http://www.receptorlight.uni-jena.de/`

[5] `https://www.uni-jena.de/`, `https://www.uni-wuerzburg.de/`

[6] `http://www.uniklinikum-jena.de/`, `http://www.ukw.de/`

[7] `http://www.ipht-jena.de/`

are complex protein molecules located in the cell membranes waiting for chemical signals. When these chemical signals are received by the receptor, they change their form and properties and communicate the signals to other parts of the cell.

Interviews with the scientists in the CRC as well as a workshop conducted to foster reproducible science[8] helped us to understand the different scientific practices followed in their experiments and their requirements of reproducibility and data management. The detailed insights from the interviews are outlined in the Section 7.2 and the excerpts from these interviews are presented in Appendix A.

There is a difference in the way of conducting experiments in different fields of science. *In Vitro* experiments are performed in a controlled environment outside of a living organism[9]. Many experiments in biology and medicine are classified in the category of *in vitro* studies. The *in silico* experiments use computational models and simulations to conduct research. In all these experiments, the components required to achieve reproducibility are data, procedures, execution environment conditions and the agents responsible for the experiment. In most cases, samples used in the *in vitro* experiments cannot be physically preserved for long but they can be digitally conserved. Conservation of an object is the process of describing and recording information related to the object so that it can be reproduced later [Pérez and Pérez-Hernández, 2015].

From the interviews, we understood that most of the scientists in fields like biology and medicine still use the conventional way of recording their data in hand-written lab notebooks. The problem in this way of storage arises when a researcher leaves the project and a new researcher joins the project. The new researcher has to get information regarding the project, previously conducted tests, analysis and results to understand the previous work. The difficulty in understanding the notes, following different approaches and standards, understanding undefined abbreviations are the few challenges faced by the new researcher. The same situation also occurs when two or more groups are working collaboratively on the same project in different locations. There could be a chance of conflict in experimental data and results. Hence, there should be a shared understanding of experimental data between the scientists so that they can reuse and analyze the results. So, it is important that the data and the results are shared and reused in an accepted way among the scientists working collaboratively in the same or different research projects.

Experiments performed by scientists can also result in different anomalies and inconsistencies due to several reasons. They can be due to some configuration of a device, property of a material or error in the procedure. In addition to device errors, it can be a human error. To detect these errors, experimental data, processes, experi-

[8] http://fusion.cs.uni-jena.de/bexis2userdevconf2017/workshop/

[9] https://mpkb.org/home/patients/assessing_literature/in_vitro_studies,Accessed on March 17, 2019.

ment environment and details of responsible persons have to be properly structured and documented. The scientists would like to track the errors and expose only those datasets which resulted in the error by querying from the large volume of data. Examples of such queries are "Which experiment used the material which was referenced in the X journal but was not verified?" or "Which dataset from the Experiment X resulted in the drop in the graph during the time period x to y?". The scientists would like to get answers to these questions for later analysis and reproducibility of their experiments. Currently, scientists could not perform this kind of analysis since the structure of the experiment is not modeled and expressed in a uniform way.

1.2 Basic Terminology

In this section, we will introduce the basic terminology used in this research work. This is necessary since some key terms used throughout the thesis are defined differently in different domains.

1.2.1 Reproducibility and Provenance

Repeatability and Reproducibility are distinct terms although they are often used interchangeably. The paper [Freire and Chirigati, 2018] provides a formal definition of a reproducible computational experiment which is defined as follows:

Definition 1.2.1. *"An experiment composed by a sequence of steps S that has been developed at time T, on environment (hardware and operating system) E, and on data D is reproducible if it can be executed with a sequence of steps S' (different or the same as S) at time $T' \geq T$, on environment E' (different or the same as E), and on data D' (different or the same as D) with consistent results."*

According to [Taylor and Kuyatt, 1994], repeatability, in the context of measurements, is getting similar or close-by results whenever the measurement is carried out under the same conditions which include the same procedure, observer, instrument, and location. Reproducibility, on the other hand, is more stringent. It refers to the capability of getting similar results whenever the measurement is carried out by an independent observer using different conditions of measurement including the method, location, or instrument.

Missier [Missier, 2016] describes that provenance plays an important part in achieving reproducibility. According to the Oxford Dictionary, provenance is defined as "*the source or origin of an object; its history or pedigree*". Provenance is the description of the process or a sequence of steps that together with the data and the parameters led to the creation of a data product [Herschel et al., 2017]. There are two forms of provenance in computational science [Freire et al., 2008]:

- **Prospective Provenance:** captures the specification of a computational task such as a script or a workflow and the steps that must be followed to generate a data product.

- **Retrospective Provenance:** captures the execution of a computational task including the steps and the environment used to derive a data product. It is a detailed log of execution of a computational task. It captures what actually happened during the execution of a computational task.

According to [Gupta, 2009], there are two types of provenance: *data provenance* and *process provenance. Data provenance* is defined as a record trail of the origin of a piece of data along with an explanation of how and why it got to the current state. While, in *process provenance*, which is especially used in business applications, an instrumented process capturing software tracks the life cycle of data generation and transformation.

Provenance helps to understand how a result is derived by examining the sequence of steps or the path taken by a scientist. The need for systematically capturing, modeling and managing provenance is prevalent across domains and applications of science. The different notions of provenance and the provenance management tools are discussed in detail in Chapter 3.

1.2.2 Semantic Web

The term “Semantic Web” coined by Tim Berners-Lee refers to the vision of the Web of Linked Data [Berners-Lee et al., 2001]. The Semantic Web is an extension of the World Wide Web to support the Web of data in addition to the Web of documents[10]. It helps people to describe the data in common formats to integrate data coming from different data stores using vocabularies. The Semantic Web technologies allow to share and reuse not only the data but also the relationships among the data across various applications. The Linked Data [Bizer et al., 2009], which is the collection of the interrelated datasets on the Web, is published and linked using Resource Description Framework (RDF) [Cyganiak et al., 2014]. RDF is used as a general method used in knowledge management applications to model or describe the resources. These resources are described in the form of *Subject-Predicate-Object*, which are also called triples. The subject describes the resource and the predicate describes the relationship between the subject and the object. A collection of RDF statements results in a labeled directed graph. RDF triples are stored in triplestores which are queried using SPARQL [Prud’hommeaux and Seaborne, 2008]. SPARQL is a query language to access and retrieve the RDF data. Apache Jena[11], RDF4J[12],

[10] `https://www.w3.org/standards/semanticweb/`

[11] `https://jena.apache.org/`

[12] `http://rdf4j.org/`

and OpenLink Virtuoso[13] are some of the frameworks for building Semantic Web applications. Describing the data in RDF makes it machine-readable so that the machines can process this data. In Linked Data, Uniform Resource Identifiers (URI)s are used to identify any object or concept [Berners-Lee et al., 2001]. Using HTTP URIs, people can look for useful information about the object.
An ontology is a common vocabulary which is used to define the concept and relationships for representing a domain [Noy et al., 2001,Studer et al., 1998]. Ontologies are used by researchers to share domain knowledge. These are developed to share a common understanding of the domain and enable the reuse of the domain knowledge. They are also used to organize knowledge helping the people and machines to communicate without ambiguity. The ontologies can be represented by using RDF. The basic format of ontology representation is RDF/XML with other alternatives provided by N3, Turtle, or JSON-LD [Lanthaler and Gütl, 2012]. The collection of technologies such as RDF, Web Ontology Language (OWL) [McGuinness et al., 2004], and SPARQL form the foundation of the Semantic Web.

1.3 Contributions

Reproducibility is not a one-button solution. The interaction of human beings in reproducibility is inevitable. Keeping this in mind, we envision our main contributions as follows:

1. **The REPRODUCE-ME Data Model and Ontology**
   We conduct interviews and meetings with scientists from different scientific disciplines to understand their experimental workflows. Based on the discussions, we study the general components of the scientific experiments required to ensure reproducibility. We study the existing provenance models and based on that, we develop the REPRODUCE-ME Data Model to represent the complete path of a scientific experiment. To share a common understanding of the scientific experiments among people and machines, we encode the REPRODUCE-ME Data Model in OWL2 Web Ontology Language. The REPRODUCE-ME ontology is built on top of the existing well structured semantic web standards. This contribution is presented in Chapter 4.

2. **Support of computational reproducibility**
   We study different computational tools and how each tool supports provenance management. The Computational notebook is one such tool which supports computational reproducibility. However, the provenance support in this tool is limited. Therefore, we develop ProvBook, an extension built on top of it to capture the experimental results along with its provenance. The provenance

[13] https://virtuoso.openlinksw.com/

of several executions of the notebooks is captured over the course of time and stored in them. Sharing the notebooks along with their provenance help to ensure computational reproducibility. This contribution is presented in Chapter 5.

3. **Provenance difference of results**
ProvBook also helps to compare and visualize the provenance difference of results over several executions of a computational notebook. The extension can be used to view the difference in input and output generated in the same or different environment by same or different experimenters. This contribution is presented in Chapter 5.

4. **Semantic representation of computational notebooks and scripts**
We extend the REPRODUCE-ME ontology to describe the provenance of computational notebooks and scripts. The computational notebooks along with its provenance information can be downloaded as RDF using ProvBook. This helps the user to share a notebook along with its provenance in RDF and also convert it back to a notebook using ProvBook. The REPRODUCE-ME ontology also allows to semantically describe the computational experiments provided by scripts along with its provenance information. The provenance information of the computational notebooks and scripts in RDF can be used in combination with the experiments that used them and help to get a track of the complete path of the scientific experiments.

5. **End-to-end provenance management of scientific experiments**
We design and develop a framework, CAESAR (**C**oll**A**borative **E**nvironment for **S**cientific **A**nalysis with **R**eproducibility) which captures, stores and queries the experimental data represented using the provenance-based semantic model. One part of the system tries to automatically capture the experimental data while the other part requires the user to manually record this information. The storage of the data also provides the ability to eventually query it. The change of the experimental metadata is also preserved to track its evolution. This contribution is presented in Chapter 6.

6. **Support of a collaborative environment**
We also support collaboration among teams and institutes in conducting reproducible research, analysis, and sharing of results with the support of CAESAR and JupyterHub. The framework provides user interface widgets and components for collaborative authoring. This contribution is presented in Chapter 6.

7. **Visualization of the complete path of a scientific experiment**
We provide two visualization approaches as part of the framework so that scientists could get an overall view of the experiment and also backtrack the

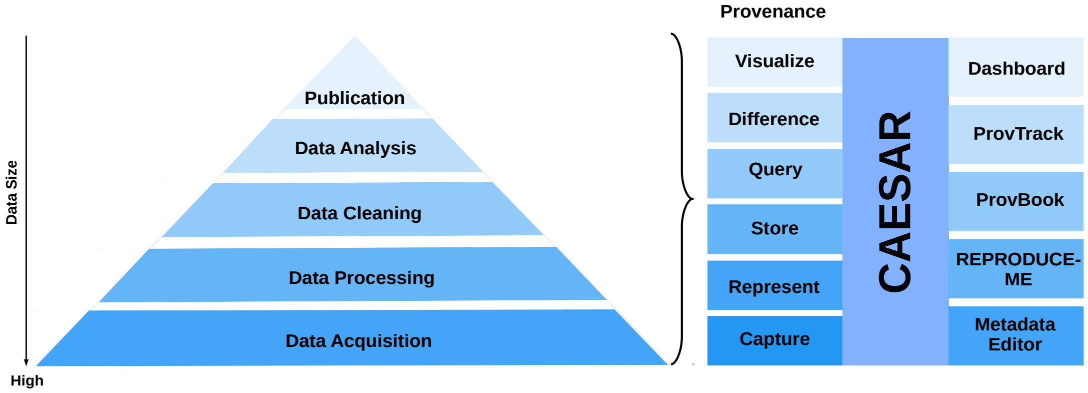


Figure 1.2: Our contributions which can be used in the different levels of a research study

steps in obtaining the results. The Dashboard and ProvTrack are the two modules in CAESAR which provides a complete overview and path of a scientific experiment respectively. This contribution is presented in Chapter 6.

Figure 1.2 shows how our contributions fit in the different levels of the research study. CAESAR can be used from the bottom level for the data management starting from the data creation to the publication level. It provides end-to-end management of experimental data by capturing, representing, storing, querying, comparing and visualizing provenance information. The REPRODUCE-ME ontology, Metadata editor, ProvBook, ProvTrack, and Dashboard are the main modules in CAESAR which help in supporting understandability, reproducibility, and reuse of scientific experiments.

1.4 Thesis Structure

This thesis is structured into the following chapters:
This chapter 1 presents the motivation for the overall research problem presented in this thesis. We briefly present our contributions to tackle the specific problem described in the introduction.
Chapter 2 presents the use case scenario and the research problems in a more formal way. Based on these challenges, we define the main hypothesis of our work. We define our goals to address the research problems. We also describe the research methodology that is followed in this thesis.
Chapter 3 presents the current state of the art in the context of reproducibility and the tools that support it. This is followed by a discussion on the gaps that exist in the current state.

Chapter 4 introduces the first contribution of our research work. It presents the REPRODUCE-ME Data Model and the ontology that is used to describe the scientific experiments and their provenance. We describe in detail the components which are important for the understandability and reproducibility of scientific experiments and how they are added in the ontology. We present the methodology that we followed in the development of the REPRODUCE-ME ontology.
Chapter 5 presents the results to support computational reproducibility. We present ProvBook tool which captures, visualizes and compares the different executions of computational notebooks. This is followed by the semantic representation of computational notebooks and scripts.
Chapter 6 presents CAESAR, a framework which is developed to capture, manage, query and visualize provenance information of scientific experiments. It presents the underlying architecture of CAESAR and discusses in detail how each phase of the provenance lifecycle is implemented. It also describes how CAESAR integrates the results from Chapter 4 and 5 to provide the complete path of a scientific experiment.
Chapter 7 presents how the evaluation was conducted for each component of our work and their results primarily focusing on the REPRODUCE-ME provenance model, ProvBook and CAESAR.
Chapter 8 concludes the thesis providing future lines of work.

1.5 Publications

Parts of the work described in this thesis have been published in peer-reviewed conferences and journals already. They are as follows:

- Samuel, S., Taubert, F., Walther, D., König-Ries, B., & Bücker, H. M. (2017). Towards reproducibility of microscopy experiments. D-Lib Magazine, 23(1/2). (Corresponds to Chapter 6)

- Samuel, S. (2017). Integrative data management for reproducibility of microscopy experiments. In Proceedings of the 14th European Semantic Web Conference, Part II (pp. 246-255). Springer, Cham. (Corresponds to Chapter 4, and 6)

- Samuel, S., & König-Ries, B. (2017). REPRODUCE-ME: Ontology-Based Data Access for Reproducibility of Microscopy Experiments. The Semantic Web: ESWC 2017 Satellite Events (pp. 17-20). Springer, Cham. (Corresponds to Chapter 4, 5, and 6)

- Samuel, S., & König-Ries, B. (2018). Combining P-Plan and the REPRODUCE-ME ontology to achieve semantic enrichment of scientific experiments using interactive notebooks. The Semantic Web: ESWC 2018 Satellite Events (pp. 126-130). Springer, Cham. (Corresponds to Chapter 5)

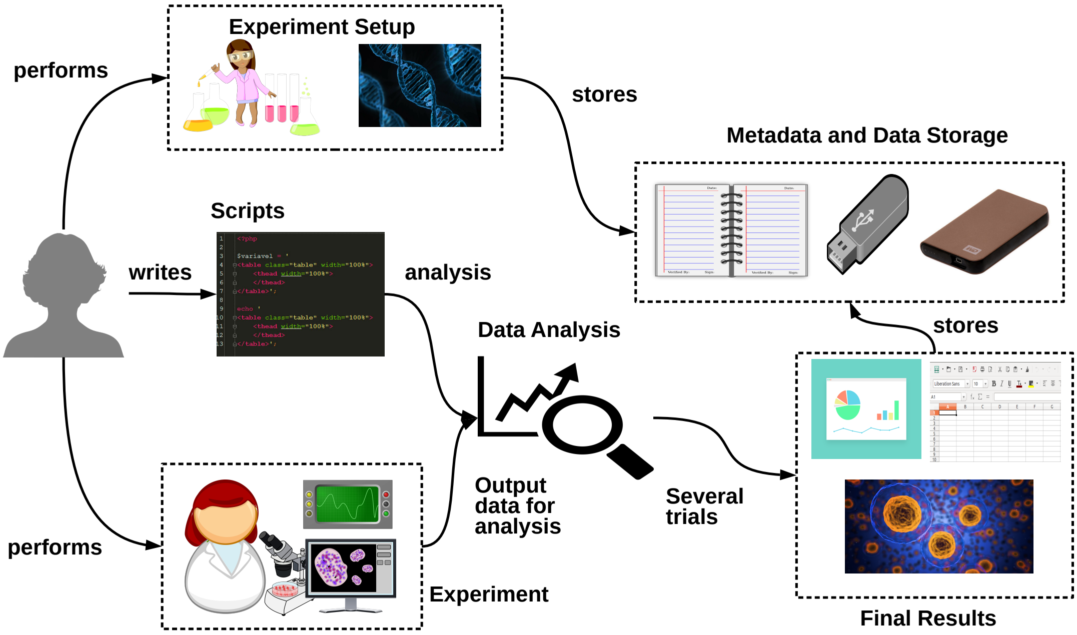
- Samuel, S., & König-Ries, B. (2018). ProvBook: Provenance-based semantic enrichment of interactive notebooks for reproducibility. Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC) 2018 (Corresponds to Chapter 5)

- Samuel, S., Groeneveld, K., Taubert, F., Walther, D., Kache, T., Langenstck, T., Knig-Ries, B., Bücker, H.M. & Biskup, C. (2018), The Story of an Experiment: A Provenance-based Semantic Approach towards Research Reproducibility. In Proceedings of the 11th International Conference Semantic Web Applications and Tools for Life Sciences, SWAT4LS 2018 (Corresponds to Chapter 4, 5, 6, and 7)

Chapter 2

Problem Statement

The main goal of this research study is to support understandability, reproducibility, and reuse of scientific experiments by supporting the researchers to represent, manage and visualize the complete path taken by scientists in performing the experiment. In order to achieve this goal, we focus our research on three key areas:
The first area of research is the **Preservation of scientific experimental data** (RA1). Experimental data preservation relates to the documentation of the metadata of an experiment and the data along with its steps and execution environment. Reproducibility of the experimental results is challenging if the path taken to obtain the result is not captured and represented. The various research practices followed in science by different researchers need to be analyzed to understand how this fundamental step of experiment preservation can ensure reproducibility.
The second area of research is the **Support of computational reproducibility** (RA2). Computational tools are widely used by the researchers in their daily work. A considerable amount of data is generated in different executions of computational experiments. The current problem is the lack of tracking of computational steps taken in an experiment as well as its computational execution environment.
The third area of research is how the first and second areas of research can be used to **capture, represent, manage and visualize the complete path** taken by a scientist in performing an experiment (RA3). The interlinking of the non-computational steps and data with the computational steps and data is a powerful method to achieve reproducibility of experiments which is missing in the current scientific practices and tools that aid reproducibility (see Chapter 3 for a detailed discussion).
In the context of these three research areas, we identify open research problems in Section 2.2 based on the use case scenario presented in Section 2.1. To address the research problems, we define the main hypothesis of our work in Section 2.3. We also discuss our goals and the requirements of the proposed solution in Section 2.4 and 2.5 respectively. The research methodology adopted for this study is presented in Section 2.6.



```
<?php

$variavel = '
<table class="table" width="100%">
    <thead width="100%">
    </thead>
</table>';

echo '
<table class="table" width="100%">
    <thead width="100%">
    </thead>
</table>';
```

Figure 2.1: Use case scenario - The experiment lifecycle of Ana

2.1 Use Case Scenario

Here we present a use case scenario showing the experimental workflow of the scientists we interviewed (see Section 1.1). Ana is a biologist who has a keen interest in doing research in biomedical science. Her research is to understand the function of the membrane receptors and she uses confocal Patch Clamp Fluorometry (cPCF) technique [Biskup et al., 2007] in her daily work. Figure 2.1 shows the experiment lifecycle performed by Ana. Before performing her experiment, she prepares all the experiment materials required in her study. She performs several steps like the preparation of samples and solutions, transfection of cells, etc. In order to do so, she refers to different publications and standard operating procedures at each step of her experiment. Several devices are used during the experiment like a microscope to capture images of the receptor cell, an electrophysiological device to generate current, etc. Each device has its own specifications. In addition to that, she configures the instruments in such a way that she could capture the image with full clarity and resolution. She needs to document the additional settings of devices configured by her besides their specifications. The execution environment of the experiment like the room temperature, humidity is also important in her case. She wants to record all the things that she has performed during the experiment including the execution environment. While she waits for the preparation step to be finished, she documents all the steps she has performed and will be going to perform in her study. Currently, she writes the important parameters used in her experiment in her laboratory notebook. Since she is part of many bigger research collaborative projects, she needs to share her experiment results in a way that other scientists can understand and reproduce.

The next step of her experiment is the analysis of the images captured by her using computational tools. The analysis is performed using a proprietary software. She writes some scripts for the further analysis of her results. She repeats the execution of scripts by experimenting with different parameters. During this experimentation, she may get negative results. However, she wants to compare these different trials of her experiment and see what resulted in the negative result. This could be useful for other scientists in her team. She then stores the raw data and the analyzed data in her external hard drive.

Before presenting her results to the other team in her collaborative project, she wants to get it reviewed by her supervisor. In order to understand, how the results were achieved, her supervisor asks for the experimental details of her study. She shows her supervisor the laboratory notebook. But she is not able to show the complete path taken by her in the experiment at one place. She needs to collect all the data and the methods she used in her experiment. Currently, her data is scattered over different devices. Some of the experimental metadata is in her laboratory notebook. The settings of the devices are stored as a proprietary file format in the software of the microscope. Some of the analyses are done using another proprietary software. The captured images are stored in her external hard disk. The scripts she used are saved in her institute computer. The trials where she got negative results are lost because she did not store all the changes that she made during the different trials of script execution with different parameters. She finds it difficult to show the link between the input data, the steps she used and the results as the data is distributed in different places and some of the data is lost while performing several trials.

Ana later presents her results in the team meeting. After the meeting, Bob, her team member, talks to Ana about her results. He is interested in her data and wants to reproduce one section of her experiment so that he could use part of her results in his own experiment. Bob goes to Ana and asks for the experiment details. He wants to understand the complete path taken by her to understand the experiment and get an overall view of her experiment. He also wants to change some parameters of her experiment and see the differences that occurred in the new result. In order to share the experimental details with Bob, she wants to direct him to one place where he could track the experiment results instead of giving him several links to various devices.

In order to tackle this problem, she wants to use an efficient experimental data management software where she could save her experimental metadata along with the other data, scripts, and results. She wants to find the connection between all the steps and the data she used and generated in her experiment in this data management platform. The platform should also be able to extract the metadata from

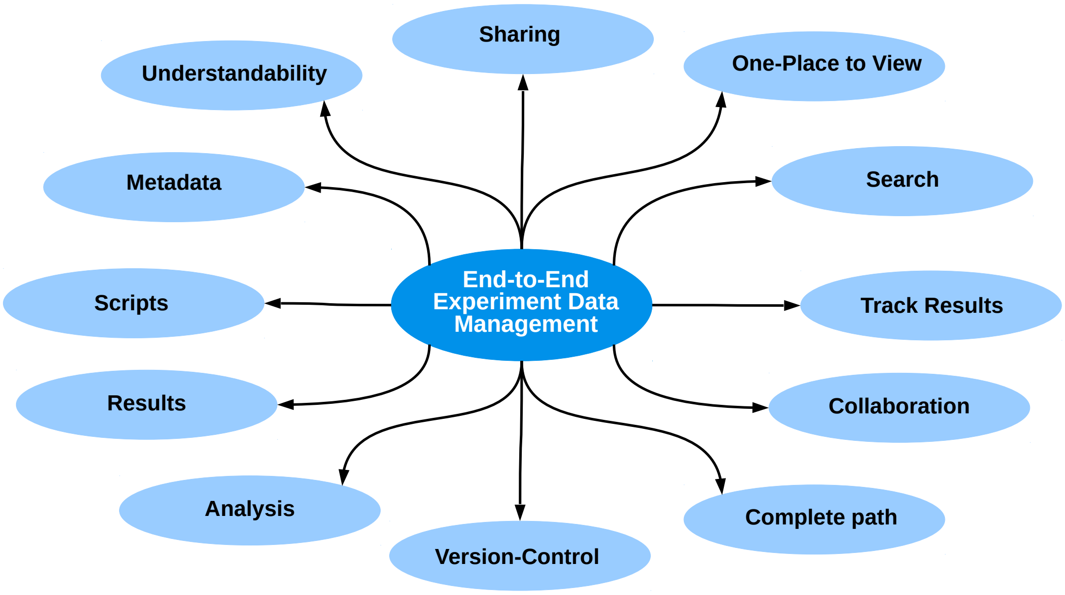


Figure 2.2: Requirements of an experimental data management platform

the images which are usually stored in different proprietary formats. This could ease her work in the documentation. She also wants that the experimental data could be written in a common format so that sharing of results becomes easier among other scientists. Another requirement is that she could work in a collaborative environment where her supervisor and other team members could make suggestions and proposals on her experiment results. Based on these thoughts and requirements of how she could improve her daily research work by using an experimental data management platform, she presents what she wants in Figure 2.2. These requirements of an experimental data management platform are also a reflection of the interviews conducted among CRC scientists (see Section 1.1). Ana wants to have this platform as a one-place to visualize the complete path of her experiment. This could help her to show how the results were achieved when asked by the other scientists of her collaborative project in the team meetings. Eventually, this would also help her in writing a publication for sharing her results to the scientific community.

2.2 Research Problems

The reproducibility crisis [Baker, 2016] shows that scientists face problem in reproducing others' results. Section 2.1 presented an example scenario which showcased the challenges faced by scientists in various aspects of understandability, reproducibility, and reuse. In their daily research work, they use and generate a lot of data through several manual and automatic steps. Several entities like instruments, execution environmental attributes, procedures, protocols, and settings are also involved in these experiments. In addition to that, there are many people involved who take several roles and responsibilities throughout these processes. There are steps that use computational resources and others that do not. It is possible that the non-computational steps have an effect on the results generated by the computational steps. Therefore, we see that there are several entities, people, activities, and steps that are linked to an experiment. However, what we often see in the publications are only the end results and some of the important methods followed to generate them.

So to understand and reproduce others' results, the question to ask here is whether this data is sufficient or do scientists need additional information. If more information is required to reproduce others' results, it is required to know what exactly are those additional components.

The research objectives of this thesis are to support understandability, reproducibility, and reuse of scientific experiments. To achieve these objectives, the important thing is to understand how the results are derived. In order to do so, it is necessary to track end-to-end provenance. Here the research problem is how to track the provenance of results because there are several challenges in doing so in cases like that mentioned in Section 2.1. We list here some of the important ones:

- Data is scattered over multiple places
- The lack of link between steps, data, people and results
- Lack of common format for sharing the results along with its provenance
- Difficulty in sharing the end-to-end provenance of results in a collaborative environment
- Loss of the data and results from different trials performed for an experiment

To address these challenges, we require a standard data model that provides a complete path taken for a scientific experiment including the computational and non-computational parts of an experiment. The data model should be able to represent the link between the results, the execution environment and the processes that generated the results of computational as well as non-computational steps of an experiment. To represent the complete path of an experiment, we need to track the provenance of the different executions in an experimental environment. We also require an approach to compare the results from the original experimenter with the results generated in a different execution environment of a computational experiment. The need for a multi-user collaborative framework which provides an integrated approach to capture, represent and visualize the provenance information of a scientific experiment along with the non-computational and computational steps should be addressed.

To address these requirements, we first need to know whether the existing provenance models are adequate for capturing provenance of a complete path of a scientific experiment. If existing, how can the existing provenance models be extended to capture provenance of complete execution of a scientific experiment? Based on our understanding of the important concepts of Semantic Web (see Section 1.2) and the advantages of using them to understand domain knowledge, we ask whether these technologies help in the understandability and reproducibility of scientific experiments.

2.3 Hypotheses

Based on the problem statement (Section 2.2), we define the main hypothesis of this thesis as follows:

> *"It is possible to capture, represent, manage and visualize a complete path taken by a scientist in an experiment including the computational and non-computational steps to derive a path towards experimental results."*

The main hypothesis can be decomposed into several sub-hypothesis.

H1 It is possible to design a data model that represents the complete path of a scientific experiment.
We divide the hypothesis H1 into three sub-parts.

H1.1 The data model is able to represent the relationship between the data, the instruments used, the settings of the instruments, the execution environment, the steps and the results of a non-computational experiment in an interoperable way.

H1.2 The data model is able to represent the relationship between the results, the execution environment and the processes that generated the results of a computational experiment in an interoperable way.

H1.3 The data model is able to represent the relationship between the computational and non-computational aspects of a scientific experiment.

H2 Semantic technologies are expressive enough to describe the complete path of a scientific experiment.

H3 An algorithmic process can be developed to track the provenance of the different executions in a computational environment.

H4 An algorithmic process can be developed to compare the results from the original experimenter with the results generated in a different execution environment of a computational experiment which will help in knowing the intermediate and negative results.

H5 A provenance-based semantic and collaborative framework to capture, represent and visualize provenance information can provide better sharing, reuse, and reproducibility of results and experimental data.

2.4 Goals

We define our goals to address the research problems. These goals will help us to verify the hypothesis and develop the contributions of this work.

Goal1 Create a conceptual model using semantic web technologies to describe a complete path of a scientific experiment.

Goal2 Design and develop a framework to keep track of the provenance of the computational experiment and its executions.

Goal3 Design and create a provenance-based semantic framework to populate this model, collecting information about the experimental data and results along with the settings and execution environment and visualize them.

2.5 Requirements

To achieve the goals of our research work (Section 2.4), we identify the functional and non-functional requirements of the proposed system.

R1 The system should be able to capture provenance of scientific experiments. Capturing provenance of scientific experiments is one of the basic as well as the challenging feature of such a system. The systems which provide automatic capture of provenance are ideal for the scientists but are difficult to develop because they should cover different user environments and also produce lot of information which is either overwhelming or does not make sense to the user [Miao and Deshpande, 2018]. User involvement is necessary to capture meaningful provenance data. There are also certain procedures like the non-computational steps of an experiment which require the involvement of users. The system should be able to capture both the computational and non-computational steps and data of a scientific experiment.

R2 The system should be able to semantically represent provenance of scientific experiments.

The system should support to semantically represent scientific experiments along with their provenance and link them to open databases. The scientific experiments along with the computational and non-computational steps should be semantically linked to provide a complete path towards their results.

R3 The system should be able to store provenance of scientific experiments.
The systems should be able to store the captured provenance of scientific experiments with both the computational and non-computational steps and data of a scientific experiment.

R4 The system should be able to query provenance of scientific experiments.
The systems should provide the facility to query the captured and stored provenance of scientific experiments with both the computational and non-computational steps and data of a scientific experiment.

R5 The system should be able to compare provenance of scientific experiments.
Comparing the difference between different executions of a scientific experiment should be supported by the system. This helps the user to see the evolution of the experimental results.

R6 The system should be able to visualize provenance of scientific experiments.
Visualization of the complete path of a scientific experiment with both the computational and non-computational steps and data is the key requirement of the system. This is important because the scientists would not be unaware of the underlying technologies of how the data is represented.

R7 The system should be able to provide a collaborative environment for sharing provenance of scientific experiments.
Collaboration among scientists in teams and projects is an important feature that needs to be supported by the system.

R8 The system should be easy to use.
In addition to the functional requirements, the system should be useful and user-friendly for scientists.

2.6 Research Methodology

The research methodology followed in this thesis is based on a standard approach where a systematic way is used to solve a research problem [Goddard and Melville, 2004]. We started with a use case driven approach as mentioned in Section 2.1. Understanding the current practices in science in performing and preserving experimental data was the first step in this work. Several fruitful meetings and discussions with scientists were conducted throughout the development of this thesis. A number of laboratory visits were also done to understand the experimental workflow of the scientists from the university. These meetings, interviews and laboratory visits pointed out the growing need of a framework for the preservation of experimental data for reproducibility and reuse in research groups in project consortiums. The recent study on the reproducibility crisis and the results from these interviews showed us the need to address this problem at the bottom level (see Figure 1.1).

A literature survey was conducted to understand the current state of the art on the approaches that aid reproducibility. The study showed that most of the works in this area are based on the Scientific Workflow Management Systems [Deelman et al., 2005] and the conservation of the scientific workflows [Liu et al., 2015]. Based on our first step in understanding scientific practices, we recognized that there are experimental workflows which do not depend or require such complex scientific workflow management systems. Many scientific workflows are based on wet lab activities and further computational analyses are performed using scripts or other software. To address such kind of workflows, we identified the research problem to link all the experimental data with its results and steps and derive a path to the results. The state of the art approaches lack to provide a connection between the results, the steps that generated them and the execution environment of the experiment in such scientific workflows (see Chapter 3).

The next step was to define the hypothesis and the goals of this work. We followed an iterative and layered approach in defining the hypothesis. The work at each layer went through the process of understanding requirements and use cases, designing the model, developing a prototype, testing and validating the prototype and finally evaluating the work. Doctoral students and scientists from several domains like biology, chemistry, computer science are involved in each phase of the work at each layer as the end-users. The methodology used at each layer is iterative, where each layer is based on the feedback received from the domain scientists. The results of each layer are used as an input for the work at the next layer. Figure 2.3 shows the research phases of this work. The contributions of this research work are based on the three goals (Goal1-Goal3).

To design a conceptual model to describe the complete path of a scientific experiment, the existing provenance models were studied (see Chapter 4). The provenance data model, PROV-O [Lebo et al., 2013] was selected because its conceptual model closely meets our requirements, and the support to interoperably extend it further for specific domain needs. Our conceptual model was developed by extending PROV-O to describe scientific experiments. To describe the steps and processes in detail, another provenance model, P-Plan [Garijo and Gil, 2012] was also selected. We used the methodology to reuse the existing standard models and extend them for this research work. To represent this conceptual model, we reviewed the use of

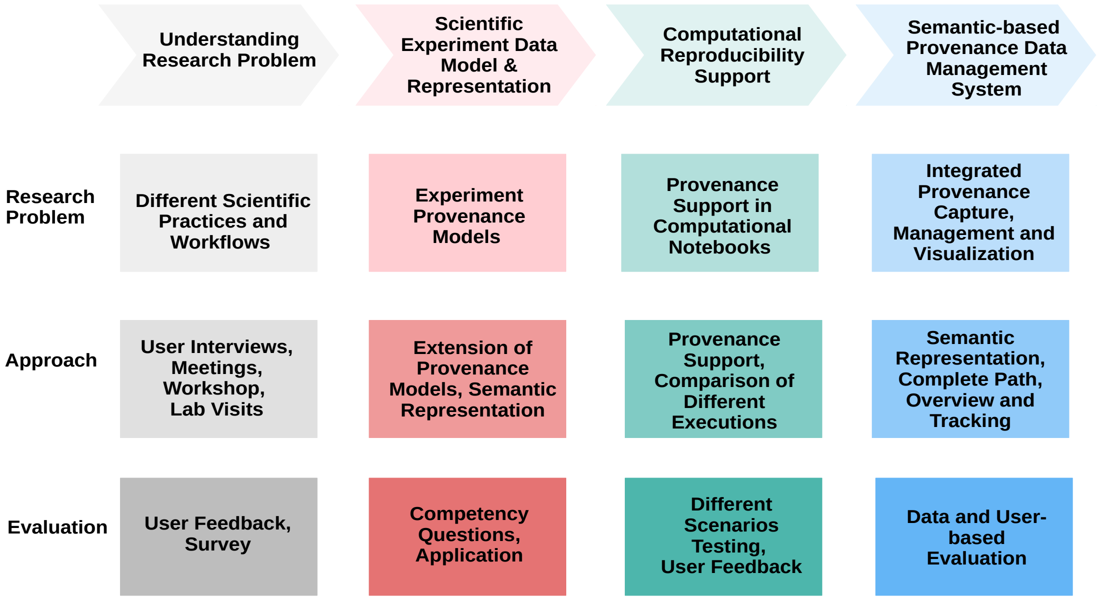

Understanding Research Problem
Scientific Experiment Data Model & Representation
Computational Reproducibility Support
Semantic-based Provenance Data Management System
Research Problem
Different Scientific Practices and Workflows
Experiment Provenance Models
Provenance Support in Computational Notebooks
Integrated Provenance Capture, Management and Visualization
Approach
User Interviews, Meetings, Workshop, Lab Visits
Extension of Provenance Models, Semantic Representation
Provenance Support, Comparison of Different Executions
Semantic Representation, Complete Path, Overview and Tracking
Evaluation
User Feedback, Survey
Competency Questions, Application
Different Scenarios Testing, User Feedback
Data and User-based Evaluation

Figure 2.3: The research phases of the thesis

semantic web technologies in describing experiments. In this phase, we designed the REPRODUCE-ME data model and the ontology to represent them [Samuel and König-Ries, 2017, Samuel, 2017].

To capture and store the experimental metadata and the data, we reviewed the existing frameworks. We limited our scope of storing provenance information for the biological domain. The extensive use of images and instruments in their experimental workflows helped us to narrow down the search to imaging-based data management systems. Two systems were used for the review and based on our requirements, OMERO [Allan et al., 2012] was selected for the underlying framework for the development of our prototype [Samuel et al., 2017]. We designed and developed the prototype to capture the provenance of scientific experiments. The use of semantic web technologies in describing scientific experiments is the key part of our work. The provenance data stored in the relational database were mapped to the ontology terms using the ontology-based data approach. At this phase, we focused on semantically describing the non-computational part of an experiment.

The next goal is to address the support of computational reproducibility. The tools that capture provenance of scripts were reviewed. The data model to represent the script provenance using ontologies was missing in the current state of the art. To describe the computational experiments, the REPRODUCE-ME ontology was extended to include this [Samuel and König-Ries, 2018a]. The extensive use and the open availability of computational notebooks motivated us to look into this direction. The computational notebooks provided rich features to run and share the experiments and results. We analyzed the computational notebooks and their structure to see what provenance information is missing and how we could extend them further to support reproducibility. The lack of tools to capture the provenance information of the executions of the computational notebooks resulted in the development of ProvBook [Samuel and König-Ries, 2018b]. The feature to compare the differences between the several executions in a notebook was also added to ProvBook. This module focused on capturing and describing the provenance of the computational part of an experiment.

The next step was to integrate the first two modules together to get an integrated approach to describe the complete path of an experiment. The non-computational and computational processes of an experiment were described and linked using the REPRODUCE-ME ontology. To help scientists get the complete picture of the experiment, visualization modules were developed [Samuel et al., 2018]. A dashboard was developed to give an overview of all the experiments that were conducted for a project. The ProvTrack module was developed to track the provenance of individual scientific experiments. To reduce the learning curve of scientists, a visual-based dashboard and ProvTrack are the entry points to our developed tool. Thanks to these approaches, the underlying technologies are transparent to the scientists.

Chapter 3

State of the Art

The research on the reproducibility of scientific results is a matter of attention in every field of science. The scientific practices followed vary across disciplines, research institutes, and teams. This results in distinct challenges with regard to supporting reproducibility. Therefore, it is vital to understand the underlying common research problems faced by scientists keeping their scientific field in mind.

There are several works which help in supporting reproducibility of results. In this chapter, we first closely look at how reproducibility is defined and the factors that are required to support it in the current state of the art. We then review the computational tools that are developed towards this purpose. This is followed by a discussion on how the experimental data is captured and represented effectively for the understandability. We survey the current state of the art and analyze how we could extend the existing works to bridge the gap that exists in this area.

A scientific experiment is represented as a dataflow composed of a sequence of computational steps where the output data of a step is used as input of another or the following step [Freire and Chirigati, 2018]. The definition of a reproducible computational experiment is given in Chapter 1 (see Definition 1.2.1) [Chirigati and Freire, 2017]. However, this definition of reproducibility focuses only on computational experiments. The steps of a scientific experiment can either be computational or non-computational. Computational steps are the ones which use computing tools to perform an activity. These tools include computers, software, scripts, etc. We consider non-computational steps as steps which do not involve computational resources. This includes activities in laboratory like preparation of samples and solutions, setting up the execution environment of the experiment, etc. Reproducing a computational step is different from a non-computational step. A computational step can be reproduced if the script or the software along with the data are provided. However, there are exceptions to that. For example, the computer programs that work with random numbers present a different challenge in the context of reproducibility. On the other hand, reproducing a non-computational step is dependent on several factors including the experimenter, the execution environment, the ex-

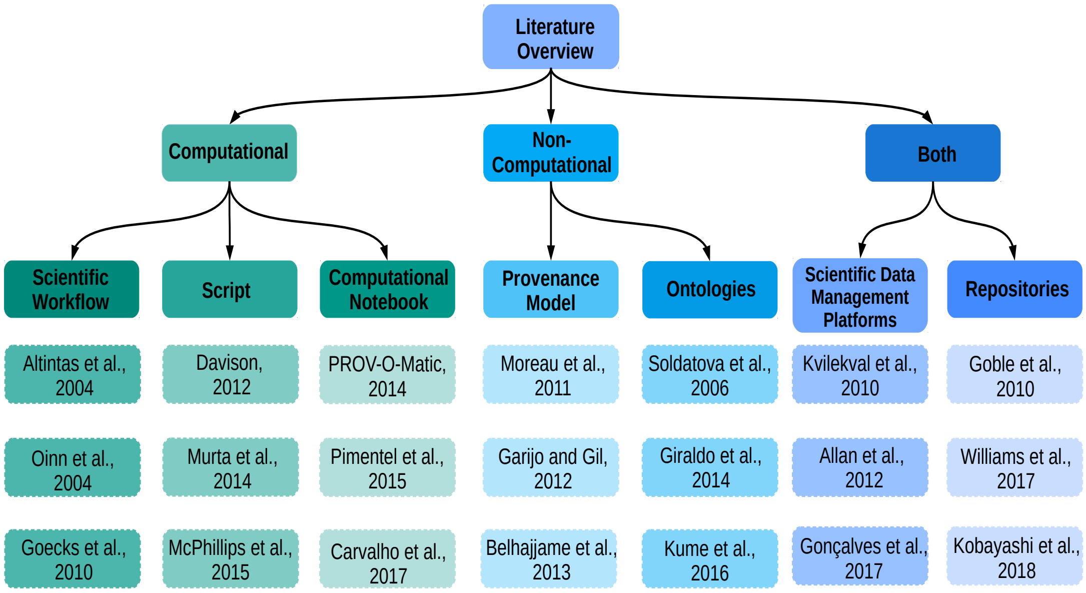


Figure 3.1: Overview of the literature survey for this research work

periment materials (e.g. animal cells or tissues), the origin of the materials (e.g. distributor of the reagents), the availability of instruments, human and machine error, etc. To reproduce a non-computational step, it is required that the step is described in detail. Kaiser [Kaiser, 2015] presents that there is a need to report every detail of the experiment including the lot number of reagents and the datasets in order to repeat an experiment. Therefore, it is important that the experiment is described in a way that helps other scientists to understand it. In order to do so, it is required to model, capture and manage the provenance of a scientific experiment in an interoperable way for the scientific community, which is still a challenge. It is essential to know what constitutes a scientific experiment and which provenance information is essential to describe this path towards the derivation of its results. Therefore, we analyze and discuss the current state of the art which covers both the computational and non-computational aspects of reproducibility of scientific experiments. Hence we categorize our literature survey into three parts:

1. Work on computational aspects of reproducibility
2. Work on non-computational aspects of reproducibility
3. Work on both computational and non-computational aspects of reproducibility

Figure 3.1 shows an overview of the literature review carried out for this research work in these categories. The literature survey first explores the computational aspect of reproducibility in Section 3.1. It analyses the current tools developed to support computational reproducibility. These tools are analyzed based on their applications of use. The three different applications that we review here are: Scientific

Workflows (Section 3.1.2.1), Scripts (Section 3.1.2.2), and Computational Notebooks (Section 3.1.2.3). The next section 3.2 reviews work on the non-computational aspect of reproducibility. This section discusses how the semantic representation of scientific experiments as linked data help towards understandability and reproducibility. It is then followed by the current state-of-the-art survey of different provenance models (Section 3.2.1). Next section reviews work on both computational and non-computational aspects of reproducibility (Section 3.3). The current state of the art is evaluated in light of the research problems (see Section 2.2) of our research work.

3.1 Work on computational aspects of reproducibility

In this section, we discuss the computational reproducibility and the related research works that support it.

The definition of a reproducible computational experiment according to [Freire and Chirigati, 2018] is stated in Section 1.2.1. To reproduce an experiment, it is essential to capture, represent and publish its provenance information. According to the definition 1.2.1, the provenance information required to reproduce an experiment includes the following details:

- A description of the input and the output data (D)
- Environmental information where the experiments are run (E)
- The steps required to run an experiment (S)

According to the Reproducibility Guide provided by the rOpenSciProject, computational reproducibility is achieved when detailed information about the code, software, hardware, and the implementation attributes are provided[1].

There are several efforts in formally defining reproducibility. The work [Moreau, 2011] provides reproducibility semantics for Open Provenance Model (OPM) [Moreau et al., 2011] graphs. The author defines *"a provenance graph as reproducible, if combined with a primitive environment, it contains enough information to be interpreted as a program (or workflow) whose execution can yield an isomorphic provenance graph"*. The reproducibility semantics provides a foundation for provenance-based reproducibility theory.

In another work, reproducibility is defined in the context of scientific workflow management systems [Liu et al., 2015]. According to [Bánáti et al., 2016], *"if and*

[1] `http://ropensci.github.io/reproducibility-guide/sections/introduction/`, Blog: Accessed on January 29, 2019.

only if every job of scientific workflow is reproducible, then the scientific workflow is reproducible".

In our work, we understand different experimental workflows to determine the provenance information that is required to reproduce an experiment and provide a formal definition of reproducibility based on that.

Full reproducibility of scientific experiments is a desirable thing but it is hard and difficult to attain. There are different levels of reproducibility that each research work tries to achieve. Freire et al. [Freire et al., 2012] present three criteria to characterize experiments based on the levels of reproducibility:

- **Depth** refers to how much information about an experiment is made available. In the current publishing environment, figures and results are included in scientific papers. But nowadays authors are also asked for the data that were used in the experiments. The higher the depth, the better is the possibility to attain reproducibility.
- **Portability** refers to whether the results can be reproduced
  - in the original environment
  - in a similar environment (i.e., similar operating systems but different hardware)
  - in a different environment (i.e., different operating systems and hardware)
- **Coverage** refers to how much of the experiments can be reproduced. For example.
  - partially reproduced experiments
  - fully reproduced experiments

  Some experiments cannot be fully reproduced because of several reasons like complexity, availability of hardware or the execution environment.

These three different criteria are independent of each other. Experiments with high coverage and depth may still not be portable. Even if the experiment has higher coverage, the lower depth can affect its reproducibility. We consider these three factors while developing the tools in our work. We also consider the factor of usability so that the user is not overwhelmed by the depth of the provenance information.

3.1.1 Provenance in Computational Experiments

Provenance can be collected at different levels from fine to coarse. The more granular the provenance information is, the more information is available for debugging and analysis. The amount of provenance information collected at each level depends on the user requirements to answer their queries. However, the size of the provenance information can grow more than the actual data [Chapman et al., 2008]. There are several developments in the past years to capture, model and manage provenance in applications. According to [Freire et al., 2008], a provenance management solution consists of three main components:

- Provenance Capture
- Provenance Model
- Provenance Management and Query

Provenance Capture
The provenance capture mechanisms collect information related to a computational task including the input data, steps, execution information, and user-defined annotations. According to [Freire et al., 2008], provenance capture mechanisms fall into three main categories: *Workflow*, *Operating System* and *Process*. Workflow-based provenance capture mechanisms collect provenance from scientific workflows in scientific workflow management systems [Sarikhani and Wendelborn, 2018]. Process-based provenance capture mechanisms require each process in a computational task to document itself. While OS-based mechanisms capture provenance at the operating system level [Frew et al., 2008,Guo and Seltzer, 2012,Muniswamy-Reddy et al., 2006]. Data and data-process dependencies are captured at the kernel level using the filesystem or user levels using the system call tracer. Capturing provenance at the operating system level is out of the scope of this thesis.

Provenance Model
Provenance Models support prospective and retrospective provenance (Section 1.2). Provenance Models differ across domain and user requirements. There are some models which try to capture the general concepts while some other target on specific use cases. For example, Vistrails [Scheidegger et al., 2008] was developed to support the visualization of exploratory computational tasks and captures the workflow evolution. Taverna [Oinn et al., 2004] was developed to support the bioinformatics workflow and thus supports ontologies from this domain. Galaxy [Goecks et al., 2010] also focuses on bioinformatics workflows.

Provenance Management and Query
A wide range of technologies ranging from the XML files to Semantic Web technologies is used to store the provenance data. Storing provenance in a filesystem gives the advantage that no other additional infrastructure is required in particular to store provenance information. Relational database is another way of storing provenance providing centralized and efficient storage. Semantic Web languages such as RDF and OWL are used to model provenance graphs thus enabling to query the

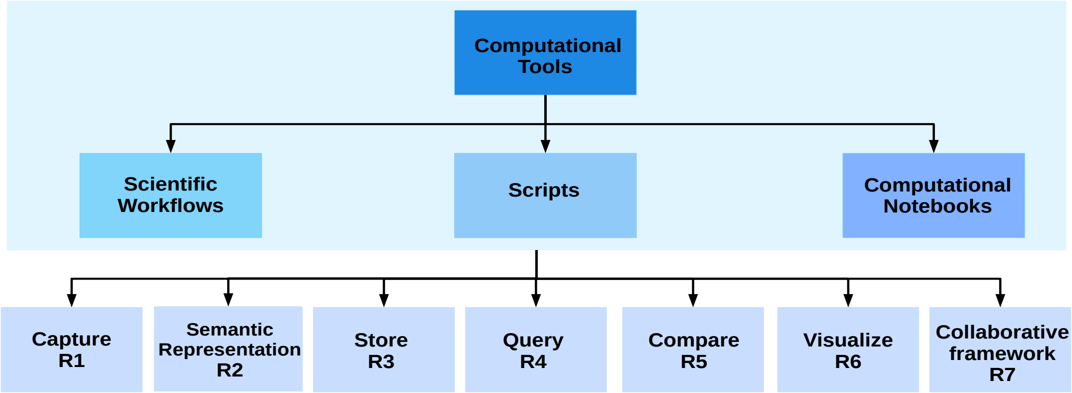


Figure 3.2: The taxonomy of computational tools.

provenance information using SPARQL.
Even though the relation between provenance and reproducibility was introduced in 2008 [Davidson and Freire, 2008], the role of provenance to support reproducibility is considered as a challenge [Missier, 2016]. The author mentions that the practical solutions where provenance is used to support reproducibility are not simply available. The author references that explaining the differences of the results from two different executions of a process using the provenance traces [Missier et al., 2016] addresses one aspect of this challenge. However, much work needs to be done to support the part of provenance in research reproducibility.

3.1.2 Reproducibility Tools

We review the tools which capture the provenance to support reproducibility based on the area of the usage:

- Scientific Workflows
- Scripts
- Computational Notebooks

Scientists use these computational tools depending on the area of their research. Each of them is widely used to perform computational experiments. Hence, it is important that the reproducibility of the results generated from these processes is ensured. Figure 3.2 shows the taxonomy of computational tools and the features used for this review. The tools that support computational reproducibility will be further analyzed and evaluated based on the functional requirements defined in Section 2.5.

3.1.2.1 Workflow Provenance

Scientific Workflow is a complex set of data processes and computations with dependencies between them [Liu et al., 2015]. These are similar to business workflows but have several challenges which are not present in the context of business workflows [Altintas et al., 2004]. Scientific workflows are more data-oriented and processing is done on large and heterogeneous computationally-intensive data. They are usually represented as a directed acyclic graph (DAG) where the nodes represent the tasks and the edges represent the dependencies between the tasks. These workflows range from a short series of tasks to long parallel tasks. They can either be simple or complex depending on the requirements of the experiments.

Scientific Workflow Management Systems (SWfMS) [Liu et al., 2015] are classical systems which help scientists to construct the scientific workflows. They help users to formally express a calculation using multi-step computational tasks [Deelman et al., 2005]. These systems guide the user to model, define, create, execute and manage the execution of scientific workflows. They help to run and manage data-intensive and complex analyses. Their aim is to enable automation, reproducibility, and sharing of experimental results. SWfMS can be useful for scientists in many ways. These include:

- An environment to design, execute and re-run their analysis
- To track the results of their scientific workflow using provenance methods
- Share the workflows with other scientists to enable reusability

There are several SWfMS developed for different use cases and domains [Altintas et al., 2004, Oinn et al., 2004, Goecks et al., 2010, Scheidegger et al., 2008, Deelman et al., 2005]. There are also well-developed systems to capture provenance, which we discuss below. We will also see how they try to support reproducibility of results in computational science.

Kepler

Kepler is a Java-based SWfMS to create and execute models for different scientific domains [Altintas et al., 2004]. It offers a GUI for workflow design using *directors* and *actors* as its main components. The provenance module is developed to capture and query workflow execution history. The provenance information is stored in a relational database and can be queried using a Java API. The provenance includes the data about actors, directors, parameters and the input/output ports in each workflow. The Kepler workflows are saved and shared by exporting them into a Kepler Archive format (KAR). This file can be shared with other scientists through email or websites.

Taverna
Taverna is an open-source SWfMS used in multiple areas like bioinformatics, chemistry, astronomy [Oinn et al., 2004]. It provides a GUI for designing workflows and also provides a command line execution of workflows. However, it does not capture the provenance of the evolution history of the workflow definition. It instead assumes that scientists can use existing systems for versioning like Git or sharing in websites like myExperiment[2]. The provenance of workflow runs is captured and stored in an internal database. The Taverna-PROV plugin allows the user to export the provenance of workflow runs in PROV-O RDF.

Galaxy
Galaxy is an open web-based SWfMS for genomic research [Goecks et al., 2010]. It provides a GUI to design workflows and share the workflow information like workflow description, input data, and provenance in a public website. It follows a directed cyclic graph approach allowing loops. Galaxy represents the workflow in JSON format. It provides *Histories* which track every change made to a workflow file.

Vistrails
Vistrails [Scheidegger et al., 2008] is another open-source SWfMS which focuses more on data exploration and visualization. It allows combining specialized libraries, resources and web services. It also provides a provenance capture and management infrastructure to track the steps and the derivation of data products. It tracks the evolution of workflows by maintaining provenance record for each workflow instance and different workflow versions.

These workflow management systems capture the provenance of workflow executions. Hence, they focus on the computational steps of an experiment and do not link to the experimental metadata. Comparison of the differences between two scientific workflows to understand the divergence of final results is currently not addressed. This is because this is a subgraph isomorphism problem which is NP-hard [Davidson et al., 2007].

Despite the availability of tools discussed above, there are currently many challenges in the context of reproducibility of scientific workflows in the workflow management systems [Cohen-Boulakia et al., 2017, Zhao et al., 2012]. The study [Zhao et al., 2012] shows that there is a *workflow decay* where nearly 80% of workflows failed to reproduce or re-run. The main reasons behind this problem as stated in the paper are improper documentation and the lack of example data. The paper [Cohen-Boulakia et al., 2017] analyzes the different workflow management systems and presents some limitations in the context of reproducibility as described below:

- Currently there are no interactive systems for the visualization and query of a large amount of provenance information.

[2] `https://www.myexperiment.org`

- Lack of interoperability between scientific workflows.
- There are larger workflow execution graphs than the workflow specifications.
- There are no approaches for the automatic annotation of tools and workflows using the terms in ontologies.
- Workflows are not citable in a manner that they can be referenced when they are reused.

The lack of interoperability between SWfMs and the steep learning curve required by the scientists are the concerns currently faced by the research community.

3.1.2.2 Script Provenance

Scripting has become a potential skill not only for computer programmers but also for scientists from different domains. Researchers globally use scripts for analysis, computation, visualization of results, etc. Scripts are easy to share with others and comparatively easy to reproduce if the data used by them are also provided. Therefore, the demand for writing scripts and sharing the results with the scientific community has increased tremendously. The complexity of the scripts and a large amount of data generated from them has increased the importance of tracking the derivation of results. Therefore, a number of tools have been developed to capture the provenance of results generated from scripts.

Provenance data can be collected from the execution of scripts at different levels of granularity. There are several tools which capture this provenance information at different levels of granularity. The tool presented by Frew et al. [Frew et al., 2008] captures provenance at the operating system level which tracks process and system calls while Tariq et al. [Tariq et al., 2012] describe a method to collect intraprocess provenance automatically. Several version management tools like Git allow developers to track the provenance of files by providing mechanisms to look at the history of versions and the ability to revert to previous versions.

There are several tools which collect provenance information from scripts at function or system level. The Sumatra [Davison, 2012] tool collects input, output, module and data dependencies from Python scripts with the version-control system. It also provides a Web-based interface to view, annotate and search provenance records. The noWorkflow tool [Murta et al., 2014] captures provenance at the function level from the scripts written in Python. It allows users to analyze the captured provenance using graph, query, and diff based analysis methods. The query-based analysis is possible by exporting the provenance data through Prolog. YesWorkflow [McPhillips et al., 2015] is another tool which collects provenance from scripts and provides many benefits of SWfMS by revealing the computational models and dataflows which are not explicit in scripts. This is possible by annotating the scripts with YesWorkflow annotations which are extracted, analyzed and presented as graphical rendering. It is a programming language-independent user-oriented tool which reveals workflow structure and dependencies from scripts based on user annotations.

The noWorkflow tool uses techniques like an abstract syntax tree, reflection, and profiling to collect different types of provenance. The paper [Murta et al., 2014] addresses the challenges of representing the environment information and determining the level of granularity of provenance information. The provenance of each execution of a script which is called trial is collected and stored so that it can be used later for other purposes by the users. They define three types of provenance:

- Definition Provenance: collects the code's structure which includes function definition, arguments, and function calls.
- Deployment Provenance: collects the execution environment of a script which includes information about the environment, operating system, and the modules used.
- Execution Provenance: collects information of what happened when the script was executed.

The noWorkflow captures the definition, deployment and execution provenance. The definition provenance is captured using the abstract syntax tree (AST) to identify the source code of each function definition. This information is associated with each execution of the script. The global variables, parameters of each function call are also captured by analyzing each function.

The deployment provenance is captured using the library provided by Python to capture about the execution environment. It captures information from the *os* library to capture operating system information, *socket* for the hostname, $platform$ for the machine architecture and the programming language environment. The noWorkflow tool uses the Python profiling API to capture the execution provenance including all the function activations of the script.

All the provenance information is stored in the SQLite database in the $.noworkflow$ directory where the script is executed. The $.noworkflow$ directory can be shared among scientists for the exchange of provenance information.

The tool provides three ways for the visualization of provenance information of the script execution. The graph-based visualization provides the summarization of the execution of script including the function activations. Figure 3.3 shows the graphical representation of a script execution using noWorkflow. The diff-based visualization provides the user the facility to compare between two different trials. It provides information on the module dependencies, environment variables, and temporal-spatial attributes. The last visualization way is providing the user to query the provenance data in Prolog.

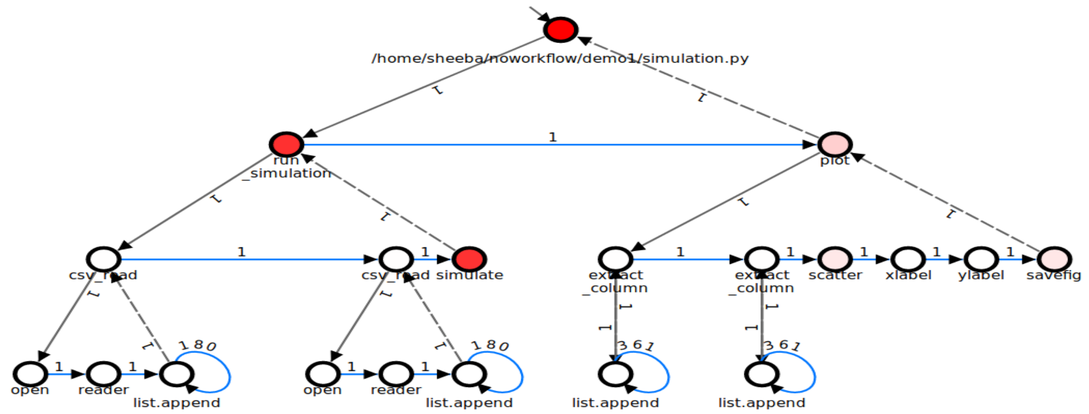


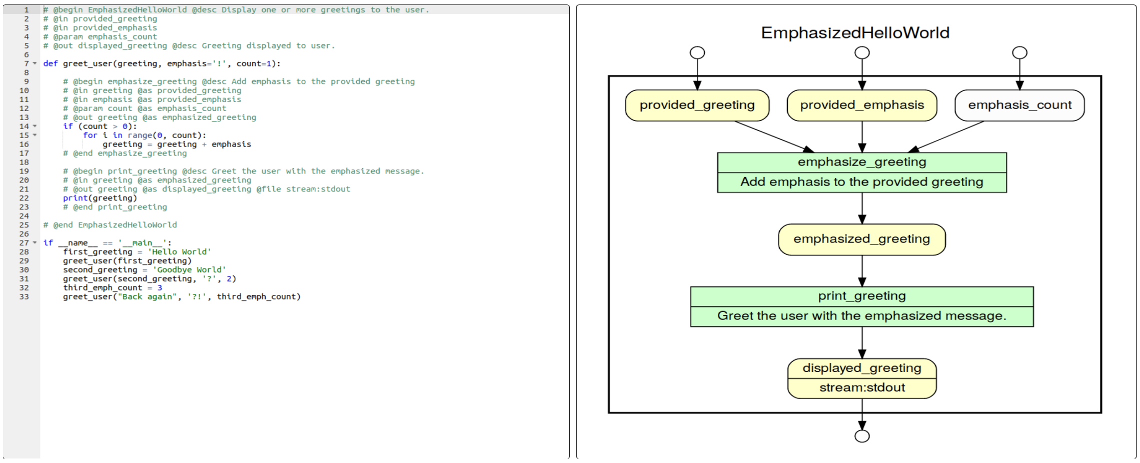
Figure 3.3: The graphical representation of a script execution using noWorkflow.

The noWorkflow tool is non-intrusive in the way that it does not require user intervention to collect provenance information. It provides the user with a fine level of granularity of provenance information thus resulting in a large amount of data. It results in cumbersome and overwhelming data. This tool can be used only for capturing provenance information from Python scripts. YesWorkflow is complementary to noWorkflow, which is language-independent and works based on user annotations [McPhillips et al., 2015]. It makes use of the benefits of scientific workflow management systems by providing the graphical visualization of provenance information in a workflow-like view. The user-annotation is done using the keywords provided by YesWorkflow. The tool provides the keywords which are based on the components of SWfMS like port, channel, workflows.

A program block of a script represents a block of code that receives input and produces output. The start and end of a program block are annotated using @*begin* and @*end* keywords. The ports of a scientific workflow are described using @*in* and @*out* keywords. A channel is described as an edge between the @*in* and @*out* ports of a scientific workflow. The extracted workflow graph from the user-based annotations is produced in GraphViz-DOT form. YesWorkflow provides a module to generate RDF representations of the YesWorkflow annotations. Figure 3.4 shows the graphical representation of a script using YesWorkflow[3].

The YesWorkflow captures only the prospective provenance and requires the user to change the script with YesWorkflow annotations. They cannot be executed by scientists. However, this could be helpful for large complex scripts to get an overall view of the script. Carvalho et al. [Carvalho et al., 2016] present a methodology to convert scripts into workflow research objects with the help of tools like YesWorkflow, Research Objects, and PROV. It is a four-step of methodology for converting scripts into reproducible Workflow Research Objects. A general abstract workflow is created from the script using the YesWorkflow user annotations. The abstract

[3] `http://try.yesworkflow.org/`



```
# @begin EmphasizedHelloWorld @desc Display one or more greetings to the user.
# @in provided_greeting
# @in provided_emphasis
# @param emphasis_count
# @out displayed_greeting @desc Greeting displayed to user.

def greet_user(greeting, emphasis='!', count=1):

    # @begin emphasize_greeting @desc Add emphasis to the provided greeting
    # @in greeting @as provided_greeting
    # @in emphasis @as provided_emphasis
    # @param count @as emphasis_count
    # @out greeting @as emphasized_greeting
    if (count > 0):
        for i in range(0, count):
            greeting = greeting + emphasis
    # @end emphasize_greeting

    # @begin print_greeting @desc Greet the user with the emphasized message.
    # @in greeting @as emphasized_greeting
    # @out greeting @as displayed_greeting @file stream:stdout
    print(greeting)
    # @end print_greeting

# @end EmphasizedHelloWorld

if __name__ == '__main__':
    first_greeting = 'Hello World'
    greet_user(first_greeting)
    second_greeting = 'Goodbye World'
    greet_user(second_greeting, '?', 2)
    third_emph_count = 3
    greet_user("Back again", '?!', third_emph_count)
```

Figure 3.4: The graphical representation of a script using YesWorkflow

workflow is a graphical representation of the script and is platform independent. This workflow is converted into a platform-specific executable workflow in their next step. The curators who are familiar with workflow and script programming are required at this stage to convert each program block in the abstract workflow to its implementation. For each program block in the abstract workflow, its associated code is copied to generate the executable workflow. The curator needs to be aware of the workflow format of the SWfMS to generate the executable workflow. The scientists then execute this new workflow and capture the provenance traces using the SWfMS. The results of the workflow are manually checked with the scripts' results and if there is a mismatch, the scientists identify the problem and re-design the workflow elements. The proposed methodology is complex for the domain scientists and requires extensive knowledge of the workflow and script programming. It also requires extensive involvement of scientists and curators in every step of the conversion.

3.1.2.3 Computational Notebook Provenance

Computational Notebooks have gained widespread adoption in recent years. These notebooks allow the data analysts to write, run and visualize the results in a single document, thus making it suitable for sharing their scientific results. The Jupyter Notebook [Kluyver et al., 2016], which was formerly known as IPython notebook, is an open-source web application which provides an interactive environment to perform data exploration, visualization, and other computational tasks. It enables the user to create documents with an interactive output. It currently supports over 100 programming languages[4] with millions of users around the world. These notebooks contain blocks of text and code which are organized as cells. The code cells contain code snippets which can be modified and executed individually and the output is displayed directly below the cell. The markdown cells contain

[4] `https://jupyter4edu.github.io/jupyter-edu-book/` Blog: Accessed on April 11, 2019

documentation of the computational processes. The cells are arranged linearly but can be moved or executed in any order. The notebook currently can be shared in different formats including HTML, PDF, and LaTeX.

In a recent study by Rule et al. [Rule et al., 2018], over 1 million publicly available notebooks on GitHub were analyzed and 15 data scientists were interviewed from different disciplines. The study was done to understand how the users actually use them and how the notebooks address the challenge of tracking and sharing the data analysis. One of the results of their study shows the need for tracking provenance. Users can over-write and re-run the cells in any order which leads to the loss of previous results. Tracking which computations and analysis have been attempted is not done automatically in these notebooks.

Tracking the provenance of results is required in such computational notebooks [Pimentel et al., 2019]. It is largely required in the trial and error experiments where it is essential to understand how exactly a final result has been achieved. It is also necessary to keep track of the experiments that have been attempted because that may benefit other scientists, even if the results are not as expected.

There are a few research works which have attempted in tracking provenance from computational notebooks. Pimentel et al. [Pimentel et al., 2015] present a mechanism to capture and analyze provenance of python scripts inside IPython-Notebooks by integrating with noWorkflow [Pimentel et al., 2017]. All the features provided by noWorkflow are therefore available in IPython notebooks. One of the limitations in this approach is that it requires the user to change the script to view the visualization. This approach allows the script to be run from IPython notebooks capturing provenance of scripts and not the provenance of notebooks. To use noworkflow in IPython notebooks, cell magic (Specific commands provided by IPython kernel[5]) is used "%%now_run". However, this approach is limited to Python scripts.

PROV-O-Matic[6] is another provenance-tracking extension for older versions of IPython Notebooks which saves the provenance traces to Linked Data file using PROV-O. Another recent approach is to convert notebooks into workflows where notebook developers need to follow a set of guidelines in writing code [Carvalho et al., 2017]. These approaches have the limitation that they require changes to scripts by the user and are limited to Python scripts. In our approach, the provenance tracking is integrated within a notebook so there is no need to change the scripts and learn a new tool. It is also easy to share the notebook along with the provenance traces of execution described as Linked Data. No work has been done to our knowledge to track the provenance of results generated from the execution of these notebooks. Hence, we will later describe how we provide an easy-to-use

[5] `https://ipython.org/`

[6] `https://github.com/Data2Semantics/prov-o-matic`, Accessed on January 29, 2019.

platform to capture and manage the provenance of computational notebook executions. We also provide an approach to make available this provenance information in an interoperable way.

3.1.3 Discussion

We discussed the computational aspect of reproducibility of scientific experiments. Table 3.2 provides an overview of the tools which support computational reproducibility. To achieve computational reproducibility, the provenance of the data, steps and execution environment are essential. Recent tools and approaches to achieve reproducibility for different kinds of workflows have been presented. SWfMS are mature systems to design and execute scientific workflows. Some of these systems attempt to capture workflow provenance. However, there are some limitations in these systems in the context of reproducibility. Workflow decay is one of them, where the workflows created are difficult for others to understand or re-run in a different environment. Improper documentation, lack of example data and execution environment results in workflow decay. The lack of interactive systems for the visualization and querying of a large amount of provenance information is one of the challenges working with these systems. Every SWfMS stores its workflows and other information in its own proprietary format which results in a lack of interoperability between scientific workflows. The Common Workflow Language is going in this direction to achieve interoperability [Amstutz et al., 2016], however, it is an ongoing work. Another area of computational reproducibility that was discussed is the scripts. Several recent tools have been introduced which capture the provenance of scripts. Some of the tools capture only the prospective provenance while others only the retrospective provenance. The noWorkflow tool captures, stores and provides a visualization and query technique for provenance management. The provenance information captured by the tool is fine-grained which results in an overload of provenance information. Additionally, this work is limited to only Python scripts. YesWorkflow tool provides the benefit of SWfMS and helps the user in providing an overview of complex scripts. However, the user has to learn YesWorkflow keywords for annotations to visualize the overview of the script. It does not provide the retrospective provenance. Another limitation of these provenance capturing tools from scripts is that they do not provide a semantic description of the provenance information. The provenance model of the scripts along with its execution is missing in these tools. In addition to this, there are some issues working in script-based environments. The lack of documentation of computational experiments along with their results and the ability to reuse parts of code present some hindrances towards reproducibility. The support of reproducible science using computational notebooks has resulted in their widespread usage. In spite of that, the provenance management in these computational notebooks is not fully supported. There are only a few tools which attempt to capture provenance of Jupyter notebooks. There is a great demand for using Jupyter notebooks irrespective of the applications used, unlike the SWfMS which are tightly coupled to certain scientific experiment types. There is no tool which captures provenance of the Jupyter notebook execution and provides the difference between the several executions of a notebook. It is also important to have a tool which is easy-to-use for every different group of users. We aim to bridge this gap by developing a tool to capture the provenance of computational notebooks and providing semantic integration of Jupyter Notebooks as well as scripts.

3.2 Work on non-computational aspects of reproducibility

To reproduce a scientific experiment, it is important that the provenance of the computational and non-computational steps are captured and described in detail. Non-computational steps do not use computing tools or resources. Hence, the provenance of such steps is mostly neither machine-controlled nor automatic. Human involvement is required to capture the provenance of non-computational steps. Hence, it is important that the provenance of these steps are clearly described. However, the recent surveys related to reproducibility has shown that the poor description of results and findability of methods and code cause irreproducible research [Baker, 2016].

Non-computational part of an experiment provides information on data, steps, execution environment, methods, and protocols. This provenance data can be represented and stored in many ways. Several approaches have been introduced to model and represent provenance to date. The approaches use a wide variety of data models ranging from Semantic Web languages (e.g. RDF, OWL) and XML stored as files to tuples in relational database model [Davidson and Freire, 2008]. The results of non-computational steps which can either be empirical or observational should also be expressed in an understandable way not only to humans but also for machines. To support interoperability and make the data machine-understandable as guided by the FAIR principles, we focus on the provenance models which rely on using shared vocabularies or ontologies.

Semantic web technologies play an important role in making this data not only machine-readable but also interoperable. The Semantic Web is an ideal environment to create knowledge bases that help interlinking scientific research data across the web. The Semantic Web languages help the scientists to capture and store the experimental metadata and interlink with other data on the web. Ontologies are a way to describe concepts and relationships and provide good contextual information [Simmhan et al., 2005]. An ontology is a formal, explicit specification of a shared conceptualization [Studer et al., 1998]. The benefits of using an ontology to represent information as stated in [Noy et al., 2001] are: 1) Sharing a common understanding of the structure of information between people and software agents. 2) Supporting the reuse of domain knowledge. 3) In making the domain assumptions elicit. 4) To help separate domain knowledge from the operational knowledge. 5) For the analysis of domain knowledge. In this section, we would look at the current provenance models and how they are used in describing scientific experiments thus supporting their understandability. We first review the provenance models and the ontologies in the context of our work.

3.2.1 Provenance Models

Several models have been introduced to represent provenance in different domains, ranging from digital humanities to biomedicine [Küster et al., 2011, Compton et al., 2012, Sahoo et al., 2019]. To have a common provenance standard, the provenance research community came forward to understand the capabilities and representation of provenance in different systems. The First Provenance Challenge [Moreau et al., 2008] was conducted to understand the similarities and differences of provenance representation in different communities. The participating teams were asked to simulate and run a well-defined Functional Magnetic Resonance Imaging Workflow. The task of this challenge was to export provenance information about the past execution of the workflow and implement and execute a set of identified queries. One of the lessons learnt from the First Provenance Challenge was that the community is missing a consistent and coherent terminology for provenance-related concepts. Since the provenance queries were considered ambiguous, a Second Provenance Challenge was conducted to address the issues from the first provenance challenge. Based on the inputs from the First and Second Provenance challenges, a consensual agreement was reached on the core representation of provenance information, the *Open Provenance Model (OPM)* [Moreau et al., 2011]. This model which was revised by a broader provenance research community was used as the model for the Third Provenance Challenge [Simmhan et al., 2011].

The OPM model was adopted by a wider part of the community after the Third Provenance Challenge. It was put forward as a data model to interchange provenance information. The W3C Provenance Incubator Group[7] was formed in 2010 with the mission to understand the requirement of provenance in different scientific domains and develop a roadmap to standardize the provenance model. This was followed by the Provenance Working Group which contributed with a family of PROV documents, a set of W3C recommendations. PROV provides a set of 8 recommendations for the interoperable interchange of provenance information among heterogenous applications [Groth and Moreau, 2013]. The key requirements, the

[7] `https://www.w3.org/2005/Incubator/prov/wiki/W3C_Provenance_Incubator_Group_Wiki`, Accessed on March 17, 2019.

principles, and the decisions that influenced the design of the PROV are discussed in [Moreau et al., 2015]. Several approaches for describing provenance semantics are discussed in the paper on the foundations of provenance on the web [Moreau, 2010]. Several ontologies were proposed before the W3C standardization effort as well. The SWAN biomedical discourse ontology [Ciccarese et al., 2008] was developed to provide the knowledge schema focusing on the authorship and attribution of personal and community organization in the area of biomedicine. It was one of the foundation models for the design of PROV along with other ontologies [Moreau et al., 2015]. The Provenir ontology [Sahoo, 2010] was developed independently of OPM and designed to model domain-specific requirements. It is an upper-level ontology consisting of three main classes: data, process, agents which are similar to the classes in PROV (Entity, Activity, Agent). However, some of these ontologies are superseded by the W3C standard, PROV [Ali and Moreau, 2013].

After the introduction of OPM and PROV, several provenance models were developed mostly focusing on scientific workflows. P-Plan [Garijo and Gil, 2012] is an ontology which extends PROV to represent the abstract scientific workflows as plans. It extends the *Plan* in PROV to track the provenance traces of plans and their past executions. It is a general purpose vocabulary to capture the main dataflow constructs and link them to the execution of a workflow. Even though it is developed to model the executions of scientific workflows, the general terms introduced in it make it possible to use in other contexts as well. This work meets our requirements and can be extended to represent scientific experiments and their executions. The other benefits of using this work are that it uses the W3C standard PROV and provides the ability to extend it further. OPMW [Garijo and Gil, 2011] is another ontology which is used to represent simple workflows with fine granularity. It extends P-Plan, PROV, and OPM. It captures the prospective and retrospective provenance of scientific workflows by linking the template, instance, and execution of the workflow. However, the paper [Missier et al., 2013] indicates that OMPW has resulted in overloading of OPM terms without introducing any additional vocabulary. They propose to use D-PROV, which is an ontology to capture the provenance traces of scientific workflow execution. This is developed in the context of DataONE [Michener et al., 2011] (Data Observation Network for Earth and hence the 'D' in D-PROV) to help scientists from DataOne to store workflow provenance traces along with the data products. It extends PROV to capture retrospective and prospective provenance of both channel and port-based scientific workflows. It also represents complex scientific workflows which include loops and optional branches. ProvONE is another ontology which is based on DataONE. It is designed to support a number of broadly used SWfMS [Cao et al., 2014]. It aims to capture the most relevant information from the computational processes in scientific workflows and provides the ability to

extend to include specificities of particular SWfMS[8].
In spite of many provenance models for representing scientific workflows, the study [Zhao et al., 2012] showed that there is a *workflow decay* based on the analysis in which nearly 80% of 92 workflows from Taverna and myExperiment[9] were failed to be executed. In order to prevent workflow decay, the approach of Research Objects along with the checklists to support workflow preservation was introduced [Belhajjame et al., 2015].
The Workflow-centric Research Objects consists of four ontologies to support aggregation of resources and domain-specific workflow requirements [Belhajjame et al., 2015]. The Research Object ontology (*ro*) is used for the description of the aggregation of resources. The Workflow Description Ontology (*wfdesc*) is used for the description of workflow specifications. The workflow provenance ontology (*wfprov*) is used for the description of the provenance traces for the execution of scientific workflows. The evolution of workflows is described using the Research Object Evolution ontology (*roevo*). These focus on descriptions of nested subworkflows as well. Even though these ontologies are used to represent scientific workflows in SWfMS, this is one of the closest work to ours. The complete path for a scientific workflow could be described using Research Objects since they represent the resources, the prospective and retrospective provenance and the evolution of workflows. Inspired by this work, we apply the idea in the context of scientific experiments.
Currently, the descriptions of workflows are described in different languages by different workflow systems. There are several models to represent scientific workflow executions but at the moment there is a lack of a standard vocabulary. There are at present hundreds of different SWfMS with more or less no interoperability between them. There are several efforts to represent workflows in a unified language. In order to avoid the lack of a standard workflow language, a project has started to overcome this barrier and it is under development. The project introduces Common Workflow Language (CWL)[10] which is a specification to describe tools and workflows to aid portability between environments [Khan et al., 2019].
From the literature survey, it is seen that most of the provenance models are developed to describe scientific workflows. The ontologies like D-PROV, ProvONE, OPMW, DataOne Ontologies are developed with the focus on modeling scientific workflows in the SWfMS. Even though our work is not directly using SWfMS, it is important to review the provenance models and how they have evolved over time to meet the requirements of systems and applications. In our approach, we focus on vocabulary which provides general provenance terms which could be used and applied to conceptualize the scientific experiments. PROV-O is a recommendation provided by the W3C group. It provides general concepts which can be used to

[8] http://purl.org/provone

[9] https://www.myexperiment.org

[10] https://www.commonwl.org/

represent and exchange provenance information in different systems and contexts. The authors of the PROV-O encourage users to extend it based on the needs of the domain. Many ontologies like D-PROV, ProvONE, DataOne, P-Plan are extended from PROV. Therefore, PROV-O is suited to conceptualize any system with provenance information. Even though the P-Plan ontology is developed to model the steps and the variables of the scientific workflows, it provides general concepts which can be used to model the steps of an experiment.

Apart from the general purpose vocabularies to model provenance, there are many ontologies which are developed to capture the requirements of individual domains. In our work, we require to model scientific experiments which consist of both computational and non-computational processes. In order to do so, we model experiments which uses light microscopy imaging techniques. Hence, we review the ontologies based on three applications:

1 Ontologies for modeling experiments

2 Ontologies for modeling light microscopy imaging experiments

3 Ontologies for modeling computational experiments

Ontologies for modeling experiments

In this section, we review the solutions and vocabularies which model experiments in general. Several models have been introduced to model the experimental metadata in different domains. The Minimum Information for Biological and Biomedical Investigation (MIBBI) [Taylor et al., 2008] is one such effort to ensure sufficient information is provided when reporting experimental data. It provides a set of minimum information checklist for the data providers to include in their experimental data in different domains. The set of such checklists are available in the MIBBI portal. The goal of this project is to promote transparency, accessibility and quality assessment of data. However, the minimum information checklists are developed independently within particular domains. This results in the redundancy of data across checklists and gets difficult to track the evolution of such checklists. This presents a difficulty for both the developers and the users of the checklists[11].

Another approach for describing experimental metadata is provided by ISAtools. The Investigation, Study Assay (ISA) is a framework for describing metadata of life sciences and biomedical experiments [Taylor et al., 2010]. It provides an abstract model which consists of three core entities to capture the experimental metadata.

- Investigation
  It describes the context of the project including title, description of the investigation and the people and publications associated with the investigation. It provides a link to the related Study of an Investigation.

[11] `https://www.force11.org/node/4660`, Accessed on March 17, 2019.

- Study
  It describes a unit of research about the metadata of the resources.

- Assay
  It describes the analytical measurements and technologies used in a study.

The abstract ISA model was originally implemented as a tabular format (ISA-Tab). Currently, it is available in two format specification: ISA-TAB and ISA-JSON. A conversion tool has been developed to transform the ISA-Tab format into RDF [González-Beltrán et al., 2014]. However, this is a software component which converts the existing ISA-Tab datasets to RDF.

BioSchemas [Gray et al., 2017] is a recent collaborative effort with the aim of making life science datasets findable using Schema.org[12] markups. It extends Schema.org to include domain-specific types like *event* and *protein*. The people in life sciences are encouraged to use markups provided by Bioschemas to include structured information on their websites. It is an ongoing development and currently provides properties only for few types.

To model scientific experiments, [Soldatova and King, 2006] presents the EXPO ontology that describes knowledge about experiment design, methodology, and results. The EXPO is extended from the upper ontology SUMO (Suggested Upper Merged Ontology) [Niles and Pease, 2001], which is proposed by the Standard Upper Ontology Working Group IEEE. This ontology is used to describe scientific experiments in general and is not tied to a specific domain. It provides more than 300 classes to describe goals, hypotheses, and results of an experiment. This work focuses more on the design aspects of an experiment and does not capture the execution environment and the execution provenance of an experiment.

The Ontology for Biomedical Investigations [Brinkman et al., 2010] is another ontology developed as a community effort to describe biomedical and clinical investigations. It is used to describe the experimental metadata in biomedical research and has been widely adopted in the biomedical domain to describe all aspects of an investigation including planning, execution, and reporting. It also reuses ontologies such as GO [Ashburner et al., 2000], Chemical Entities of Biological Interest (ChEBI) [Degtyarenko et al., 2007] and Phenotype Attribute, and Trait Ontology (PATO). Even though we do not directly use this ontology, it is used to annotate documents by scientists in our platform for capturing experimental metadata.

Several approaches have emerged to describe the experimental protocols in life-sciences[13] [Giraldo et al., 2014]. SMART Protocols (SP) is an ontology-based approach to represent experimental protocols [Giraldo et al., 2014]. The elements of SP are extracted from the analysis of 175 protocols. It extends P-Plan to represent the executable aspects of the protocol and other ontologies like EXPO, OBI

[12] `https://schema.org/`

[13] `http://autoprotocol.org/`, `https://www.protocols.io/`

for the biomedical domain knowledge. This ontology is composed of two modules: SP-document and SP-workflow. SP-document models experiments protocols as a document while SP-workflow models the protocols as a workflow. The focus of this ontology is only on the semantic representation of experiment protocols.

Ontologies such as EXPO, OBI, SWAN/SIOC provide vocabularies that allow the description of experiments and the resources that are used within them. However, they do not use the standard PROV model which prevents the interoperability of the collected data.

Another approach towards representing provenance information is using Nanopublication [Groth et al., 2010]. They are the smallest unit to publish information including the assertion, provenance, and publication information. An assertion is used to describe the relationship between two concepts and provenance provides the context of the assertation. The publication information gives the authoring and attribution data of the assertion and provenance as a whole.

Ontologies for modeling light microscopy imaging

One of our research areas is to capture the execution environment of an experiment in the context of light microscopy imaging. To describe the imaging experiments, it is important to describe how images are obtained and which instruments are used for their acquisition. Therefore, we review the works which focus on ontologies to describe light microscopy imaging experiments. A closely related work [Kume et al., 2016] presents the development of an Ontology for an Integrated Image Analysis Platform to enable Global Sharing of Microscopy Imaging Data. The authors aim to build an ontology to describe imaging metadata for the optical and electron microscopy images. They construct a Resource Description Framework (RDF) schema from the Open Microscopy Environment (OME) [Allan et al., 2012] data model. Even though there is a small overlap of their work with ours on imaging metadata, the use of PROV to represent the imaging metadata in our work provides additional benefit. Jupp et al. [Jupp et al., 2016] present the Cellular Microscopy Phenotype Ontology (CMPO) which is a species-neutral ontology for describing phenotypic observations relating to a whole cell, cellular components, cellular processes, and cell populations. This work focuses more on cell-level properties.

Ontologies for modeling computational experiments

We review the ontologies which model computational experiments in particular script and computational notebook execution. Function Ontology [Meester et al., 2016] is one approach which is developed to semantically declare and describe functions. The ontology provides concepts for Function, Problem, Algorithm, Parameter, Output, and Execution. However, it does not capture the dependencies between execution, modules, and files. Software Ontology (SWO) [Malone et al., 2014] provides a description of the software in general. It models the data, the version and the license used by the software. The work [Pérez and Pérez-Hernández, 2015] describes the infrastructure-approach of an experiment by introducing WICUS ontology but limits to describe only the computation resources like software configuration. Currently, there is no approach to model the computational notebooks and the provenance of their executions.

3.3 Work on both computational and non-computational aspects of reproducibility

Scientific data management plays a key role in knowledge discovery, data integration, and reuse. The prerequisite for good data management is provided by the FAIR principles [Wilkinson et al., 2016]. Humans are capable of understanding semantics which makes it easier for us to identify and interpret data. But it is difficult for us to act at a high speed on complex and large datasets. While machines are capable of handling data at a larger and faster scale, they are not able to understand the semantics of the data. Therefore, the guideline for FAIRness is proposed for both machines as well as humans. One of the principles of FAIR is to make the data interoperable by making it machine-readable. The data objects are interoperable "*only if the data is machine-actionable, utilizes shared vocabularies or ontologies and the data within the object should be syntactically parseable and semantically machine-accessible*" [Wilkinson et al., 2016]. One of the benefits of machine-readable data is tracking of provenance records.

Digital preservation helps in ensuring long-term data access in the present era of ever-changing technologies and research. Preservation of digital objects is studied for long in the digital preservation community. Some works give more importance to software and business process conservation [Mayer et al., 2012], while other works focus on scientific workflow preservation [Belhajjame et al., 2015]. There are also works which provide the infrastructure to support the execution of workflows. The packaging tools like Reprozip [Chirigati et al., 2013] and Docker [Boettiger, 2015] help user to create packages that include all dependencies to reproduce a computational experiment or a workflow. The tool Reprozip records workflow of command-line executions and creates packages which can be used to rerun and verify the results. However, Reprozip does not capture the evolution of workflows and uses proprietary language for workflow descriptions.

We focus our approach more towards the data management solutions for scientific data including images. The paper [Eliceiri et al., 2012] provides a list of biological imaging software tools. It presents two open-source image database. We reviewed these two imaging database management platforms: BisQue [Kvilekval et al., 2010] and OMERO [Allan et al., 2012]. The Bio-Image Semantic Query User Environment (BisQue) is an open source, server-based software system that can store, display and analyze images. The stored images can be accessed through a web interface or by using API. It is being developed and maintained by a small team at UCSB. They have two releases per year schedule. The platform uses the Bio-formats[14], OpenSlide[15], and ImarisConvert[16] to support over 240 file formats.

OMERO [Allan et al., 2012] is another open source data management platform for imaging metadata primarily for experimental biology. The OMERO software platform is developed by the Open Microscopy Environment (OME) which is a collaborative consortium responsible for producing open specifications and tools to enable open-access of image data. Its plugin architecture provides a rich set of features including analyzing and modifying images. It supports over 140 image file formats using BIO-Formats [Linkert et al., 2010]. OMERO has a very active development community ensuring a continued effort to improve the system, with everybody being able to contribute. It has also a well-documented API to write own tools and the ability to extend the web interface with plugins. It also profits from a faster release cycle. OMEROs ICE (ZeroCs Internet Communications Engine[17])-based framework is demonstrated to be scalable to very large multi-terabyte datasets across applications. The performance and the scalability while handling large heterogeneous data are important criteria in biological applications.

RIKEN [Kobayashi et al., 2018] is a meta-database platform for life-sciences. It provides datasets of genomes and phenomes of different species as well as sequence and image data. It also provides a SPARQL endpoint, a web interface for data input and an RDF converter tool.

A general approach to document experimental metadata is provided by the CEDAR workbench [Gonçalves et al., 2017]. It is a metadata repository with a web-based tool which helps users to create metadata templates and fill in the metadata using those templates. The metadata is available in JSON, JSON-LD and RDF formats. The main features of the CEDAR workbench include the Template Designer, BioPortal Lookup Service, Intelligent Authoring and Collaboration. The BioPortal Lookup Service Module in CEDAR helps the user to annotate the template using the ontology terms. The Intelligent Authoring module helps to decrease the metadata authoring time by recommending values based on the context-sensitive suggestions. It also provides REST API to export the metadata and the templates to other systems. This work is developed parallel to this thesis. One part of our work is to provide a metadata editor which overlaps with this work. The ability to query and visualize the end-to-end provenance of scientific experiments is missing.

The myExperiment [Goble et al., 2010] is a social networking environment for sharing bioinformatics workflows. Since its release from 2007, it has around 3900 workflows mainly Taverna. The workflows and the supporting files can be bundled together as

[14] `https://www.openmicroscopy.org/bio-formats/`

[15] `https://openslide.org/`

[16] `http://www.bitplane.com/`

[17] `http://www.zeroc.com`

Solution	Category	Purpose
OPM [Moreau et al., 2011]	Provenance Model	Model Scientific Workflows
PROV-O [Lebo et al., 2013]	Provenance Model	General-Purpose ontology to model Entities, Activities and Agents
P-Plan [Garijo and Gil, 2012]	Provenance Model	Model Scientific Workflows with plans and their execution
Provenir [Sahoo, 2010]	Provenance Model	Model Scientific Workflows
OPMW [Garijo and Gil, 2011]	Provenance Model	Model Scientific Workflows
D-PROV [Missier et al., 2013]	Provenance Model	Model Scientific Workflows
Research Objects [Belhajjame et al., 2015]	Provenance Model	Model Scientific Workflows with the aggregation of resources
EXPO [Soldatova and King, 2006]	Provenance Model	Model Scientific Experiments
OMERO [Allan et al., 2012]	Experimental Data Preservation	Image Database
BisQue [Kvilekval et al., 2010]	Experimental Data Preservation	Image Database

Table 3.1: Overview of the solutions for describing scientific experiments

packs so that other users can download it together. It also provides collaborative support allowing users to create and join groups. However, the difficulty in reusing of other scientist's workflow has been a major concern [Zhao et al., 2012] of this environment.

3.4 Discussion

In Section 3.2, the need for describing scientific experiments with Semantic Web technologies has been discussed. Table 3.1 provides an overview of the solutions in the context of the non-computational aspect of reproducibility. Several provenance models described using ontologies have been presented to suit for different domains of applications. After many discussions and provenance challenges, a standard is developed to capture the provenance information irrespective of the domain. Parallel to the development of this Open Provenance Model, several other ontologies like Provenir were also developed with the same aim. The W3C Provenance Working Group developed a family of documents, PROV, which became the standard model for provenance information. Since it is an upper-level ontology, it is essential to capture the provenance information in detail based on the application of use. Several approaches have been introduced to describe provenance information of scientific workflows. The workflow-centric Research Objects have been widely used to support the aggregation of resources. Ontologies like ProvONE, D-PROV are used to represent the computational processes of scientific workflows. There are only a few models which capture the execution environment of workflows. The WICUS ontology has targeted for the conservation of scientific workflows. However, there are few ontologies which capture the provenance information of scientific experiments like EXPO, SWAN. These have the limitation that they do not extend the PROV model hence resulting in the lack of interoperability.

Several approaches for provenance management of scientific experiments have also been discussed. One of our requirements is the data management of microscopy images. OMERO and BisQue are the two closest approaches which meet our requirements. We reviewed other solutions in the context of scientific data management. These solutions either focus on providing data management support of non-computational processes or management support of scientific workflows. But these solutions do not directly provide the features to support our goals. There exists a gap in them as they do not provide the feature to fully capture, represent and visualize the complete path of a scientific experiment. Hence, it is important that they are extended to support our goals and at the same time reuse their rich features.

3.5 Summary

In this chapter, we have discussed the most relevant approaches on computational reproducibility and provenance models. We have explained the general concepts which are used throughout this thesis. We then categorized existing research work into three: (1) Computational aspect (2) Non-computational aspect and (3) Both computational and non-computational aspect of reproducibility. We have analyzed the current state of the art in these areas and discussed the limitations of the approaches. Table 3.2 shows the overview of the tools used in the literature survey evaluated against the requirements R1-R6.

In Section 3.1, we discussed the approaches which support and enable computational reproducibility. We have analyzed the approaches based on the applications of usage. SWfMS play a major role in the creation and execution of scientific workflows. They help to automate the data processing steps and can repeat steps with new data. These systems are developed either for a general or specific purpose. Some of the systems provide provenance capture and management infrastructure [Scheidegger et al., 2008, Goecks et al., 2010]. While some systems [Oinn et al., 2004] use external tools like Git, myExperiment for tracking provenance. However, there are a couple of limitations of these systems in general. The paper [Spjuth et al., 2015] presents the experiences with workflows in bioinformatics. The standardization of sharing data with other applications and more effort because of the complexity of the systems are some of the challenges faced by scientists using these systems. Another challenge is the workflow decay [Zhao et al., 2012] where the existing workflows are difficult to reuse because of insufficient documentation and examples. The scientific workflows are helpful in automating complex tasks but they do not provide facility to include documentation of experiments.

Scripts are widely adopted by scientists because of their simplicity and the power to perform an analysis. The difficulty to understand and reuse others' code are also considered as challenges of scripting in the context of reproducibility. Therefore several approaches are developed recently to address this issue. Jupyter notebooks [Kluyver et al., 2016], YesWorkflow [McPhillips et al., 2015], and noWorkflow [Murta et al.,

Application Area	Tool	Computational Experiment						Non-Computational Experiment					
		R1	R2	R3	R4	R5	R6	R1	R2	R3	R4	R5	R6
Scientific Workflows	Kepler [Altintas et al., 2004]	✓	●	✓	✓	x	✓	x	x	x	x	x	x
	Taverna [Oinn et al., 2004]	✓	✓	✓	✓	x	✓	x	x	x	x	x	x
	Galaxy [Goecks et al., 2010]	✓	x	✓	✓	●	✓	x	x	x	x	x	x
	Vistrails [Scheidegger et al., 2008]	✓	x	✓	✓	●	✓	x	x	x	x	x	x
Scripts	Sumatra [Davison, 2012]	✓	x	✓	✓	✓	✓	x	x	x	x	x	x
	noWorkflow [Murta et al., 2014]	✓	x	✓	✓	✓	✓	x	x	x	x	x	x
	YesWorkflow [McPhillips et al., 2015]	✓	●	✓	✓	x	✓	x	x	x	x	x	x
Computational Notebooks	Jupyter Notebooks [Kluyver et al., 2016]	●	x	●	x	●	●	x	x	x	x	x	x
	[Pimentel et al., 2015]	●	x	●	●	●	●	x	x	x	x	x	x
	PROV-O Matic	●	✓	✓	✓	✓	✓	x	x	x	x	x	x
Scientific Data Management Platforms	OMERO [Allan et al., 2012]	●	x	●	●	x	✓	x	✓	x	✓	x	✓
	CEDAR [Gonçalves et al., 2017]	x	x	x	x	x	x	✓	✓	✓	✓	x	✓
	RIKEN [Kobayashi et al., 2018]	x	x	x	x	x	x	✓	✓	✓	✓	x	✓
	myExperiment [Goble et al., 2010]	✓	✓	✓	✓	x	✓	●	●	●	●	●	●

Table 3.2: Overview of the tools used in the literature survey evaluated against the requirements R1-R6. ✓denotes full support for the requirement, ● denotes limited support and x denotes no support for the requirement.

2014] are some of the approaches to aid understandability and reproducibility of scripts. Using YesWorkflow, the user can capture the prospective provenance using user-provided tags. However, it does not capture the retrospective provenance. The noWorkflow tool is language-dependent and provides very fine-detailed provenance which can be cumbersome for users. Computational notebooks provide the benefit of reuse and sharing of scripts along with the results and documentation. These notebooks are gaining wide-spread adoption among scientists from every domain. Even though it is a powerful tool aiding reproducibility, it lacks provenance traces of their execution. Computational reproducibility supported by these notebooks combined with the facility of tracking provenance can help to increase the understandability and simplify reuse of experiments. In our approach, we bridge this gap by including provenance management feature in computational notebooks.

In Section 3.2, we discussed the non-computational aspect of reproducibility giving importance to the preservation of experimental data using semantic web technologies. The provenance model developed by a W3C working group is a starting point to develop domain-specific models. Even though there are many other ontologies developed in parallel to PROV-O which serves the same purpose, PROV-O is widely used since it is a W3C recommendation and supports interoperability. We have also discussed some ontologies developed to model scientific experiments. EXPO [Soldatova and King, 2006] is developed to model how experiments are designed. OBI [Brinkman et al., 2010] provides classes for the annotation of biomedical investigations. Other approaches like SMART Protocols, Bioschemas, SWAN address a particular requirement (for example, experiment protocol, event, etc.). Some of these models also do not use PROV-O which is the foundation of our work. The prospective and retrospective provenance of experiments could not be modeled using these ontologies which we see as a limitation.

To describe a complete path of an experiment, it is essential that the computational and non-computational steps are expressed in a standard way. From the literature survey, we could see that the interlink between the computational and non-computational processes of an experiment is missing. There were very few approaches which focused both the computational and non-computational aspects of a scientific experiment for reproducibility. Some ontologies were developed to capture the execution infrastructure of experiments in the context of SWfMS [Pérez and Pérez-Hernández, 2015]. There were no approaches which semantically described the retrospective provenance of a script execution. One of our main aims is to fill the gap and devise a mechanism to establish a connection between computational and non-computational steps to better understand and reproduce a scientific experiment. The concept of scientific workflows to capture provenance of each step with input and parameters is applied in our research by semantically modeling the experiments. Section 3.1 presents several approaches to semantically represent scientific experiments. The approaches proposed are very diverse. However, they do not capture the complete execution path of a scientific experiment. There are several approaches which semantically model the workflow execution. But there are no approaches that provide a semantic model of a computational model of script execution. We also reviewed the provenance management systems for scientific data including images. OMERO provides rich features for imaging datasets and captures the image metadata. Another approach uses a semantic-based technique for capturing metadata of experiments which is developed in parallel with this thesis. It provides a metadata editor with templates which uses intelligent authoring with the help of ontologies [Gonçalves et al., 2017]. However, it lacks several other features required for end-to-end provenance management of scientific experiments.

There is a lack of tools which interlink the data, the steps and the results from both the computational and non-computational processes of a scientific experiment. The hypothesis of our work shows the need for end-to-end provenance management of experiments. From the literature survey, it is seen that such an approach is missing. In the following chapters, we present our approach to support reproducibility and understandability of scientific experiments.

Chapter 4

The REPRODUCE-ME Data Model and Ontology

A provenance data model is important to represent provenance information of any data object so that the data can be exchanged interoperably between systems and applications [Moreau et al., 2011]. It is also essential that the model integrates domain semantics by including domain-specific knowledge to meet requirements of users and applications in building a provenance infrastructure [Sahoo et al., 2008]. Considering these two pieces of information, we envision to develop a data model which provides provenance of scientific experiments along with domain semantics. Towards this goal, we present a conceptual data model using semantic web technologies to represent a complete path of a scientific experiment including the computational and non-computational steps to track the provenance of results (Section 2.3). To do so, it is necessary to figure out which elements are essential to represent the end-to-end provenance of scientific experiments and categorize them based on the importance for their reproducibility. For that, we first define what reproducibility means in our context. Based on this, we present the REPRODUCE-ME Data Model (REPRODUCE-ME DM) to represent the complete path of a scientific experiment which takes into account its computational and non-computational aspects. The literature survey (Chapter 3) pointed out the need for extending the existing provenance models to represent and capture this end-to-end provenance. So our model extends the existing models and standards to make our work reusable and interoperable.

This chapter first presents our definitions of several important terms used in this research work (Section 4.1). We understand the requirements for the reproducibility of experiments from the scientists' perspective in the form of competency questions in Section 4.2. This is followed by studying the current provenance models which inspired our work (Section 4.3). In Section 4.4, we introduce the REPRODUCE-ME Data Model. We present our model represented using semantic web technologies and the development phases of the REPRODUCE-ME ontology in Section 4.5. We conclude with the summary of this chapter in Section 4.6. Parts of the results of this chapter have been published in [Samuel and König-Ries, 2017].

4.1 Definitions

In Chapter 3, we reviewed the current state-of-the-art definitions of reproducibility (Section 3.1). Inspired by the definitions [Freire and Chirigati, 2018, Taylor and Kuyatt, 1994], we precisely define the following terms which we will use throughout this thesis in the context of our research work.

Definition 4.1.1. Scientific Experiment: A scientific experiment **E** is a set of computational steps **CS** and non-computational steps **NCS** performed in an order **O** at a time **T** by agents **A** using data **D**, standardized procedures **SP**, and settings **S** in an execution environment **EE** generating results **R** to achieve goals **G** by validating or refuting the hypothesis **H**.

Definition 4.1.2. Computational Step: A computational step **CS** is a step performed using computational agents or resources like computer, software, script, etc.

Definition 4.1.3. Non-computational Step: A non-computational step **NCS** is a step performed without using any computational agents or resources.

Definition 4.1.4. Reproducibility: A scientific experiment **E** composed of computational steps **CS** and non-computational steps **NCS** performed in an order **O** at a point in time **T** by agents **A** in an execution environment **EE** with data **D** and settings **S** is said to be reproducible if the experiment can be performed to get the same or similar (close-by) results by making variations in the original experiment **E**. The variations can be done in one or more of the following variables:

- Computational steps **CS**
- Non-Computational steps **NCS**
- Data **D**
- Settings **S**
- Execution environment **EE**
- Agents **A**
- Order of execution **O**
- Time **T**

Definition 4.1.5. Repeatability: A scientific experiment **E** composed of computational steps **CS** and non-computational steps **NCS** performed in an order **O** at a point in time **T** by agents **A** in an execution environment **EE** with data **D** and settings **S** is said to be repeatable if the experiment can be performed with the same conditions of the original experiment **E** to get the exact results. The conditions which must remain same are:

- Computational steps **CS**
- Non-Computational steps **NCS**
- Data **D**
- Settings **S**
- Execution environment **EE**
- Agents **A**
- Order of execution **O**

Definition 4.1.6. Reuse: A scientific experiment **E** is said to be reused if the experiment along with the data **D** and results **R** are used by a possibly different experimenter $\mathbf{A}'$ in a possibly different execution environment $\mathbf{EE}'$ but with a same or different goal $\mathbf{G}'$.

Definition 4.1.7. Understandability: A scientific experiment **E** is said to be understandable when enough information is presented to comprehend the data **D** and results **R** of the experiment by a possibly different agent $\mathbf{A}'$.

A scientific experiment, for example, which involves understanding the functions of membrane receptors, consists of several computational and non-computational steps. These steps can either be manual or automatic. Some steps use computing resources and others do not. Wet lab activities, field work, manual surveys, interviews, are some examples of non-computational steps. The wet lab activities would include preparation of specimens, solutions, setting up devices, etc. These steps follow a method, procedure or protocol (Different terms are used in different fields). The computational steps include the analysis of images using software, scripts, or computational notebooks, generating graphs for data exploration, etc. Another important thing is the order of execution. The order of performing the steps can affect the final result. For example, the cells in a computational notebook can be executed in any order which affects the final result. In order to reproduce the experiment, it is important that the computational and non-computational steps are reproducible.

Table 4.1 and 4.2 show the different cases of reproducibility and repeatability based

Variable	**Initial Experiment**	**Reproducible Experiment**				
	T	T_1	T_2	T_3	T_4	T_5
Goal	G	G	G	G	G	G′
Data	D	D′	D′	D	D	D
Code	C	C′	C	C′	C	C
Agent	A	A′	A	A	A′	A
Execution Environment	EE	EE′	EE	EE′	EE	EE
Settings	S	S′	S	S	S	S
Computational Step	CS	CS′	CS	CS′	CS	CS
Non-Computational Step	NCS	NCS′	NCS	NCS′	NCS	NCS
Order of Execution	O	O′	O	O	O′	O

Table 4.1: Reproducibility Matrix: Different cases of Reproducibility. The symbol ′ denotes change in the variable.

Variable	**Initial Experiment**	**Repeatable Experiment**				
	T	T_1	T_2	T_3	T_4	T_5
Goal	G	G	G	G	G	G
Data	D	D	D	D	D	D
Code	C	C	C	C	C	C
Agent	A	A	A	A	A	A
Execution Environment	EE	EE	EE	EE	EE	EE
Settings	S	S	S	S	S	S
Computational Step	CS	CS	CS	CS	CS	CS
Non-Computational Step	NCS	NCS	NCS	NCS	NCS	NCS
Order of Execution	O	O	O	O	O	O

Table 4.2: Repeatability Matrix: Different cases of Repeatability. The symbol ′ denotes change in the variable.

on the elements of a scientific experiment respectively. Each variable is represented using a symbol. The variables in the initial experiment remain the same. In Table 4.1, when the initial experiment is reproduced, there are several permutations where the variables could be changed. We show only some of the permutations of the reproducible experiment when changing the time **T** variable (T_1, T_2, T_3, T_4, T_5). For example, at time T_1, the goal of the reproducible experiment remains the same, while changes are made in the other variables. Whereas at time T_5, the goal of the reproducible experiment is different, while the data, steps and other variable remain the same. Table 4.2 presents the repeatability matrix where we show the state of variables when changing the time **T** variable (T_1, T_2, T_3, T_4, T_5). In the case of repeatability, none of the variables change. This is done to get the exact results of the initial experiment. The aim of repeating an experiment is to verify the results of the original experiment. While, in the case of reproducibility, the aim of reproducing an experiment is to see whether the results are consistent with the results from the original experiment. As shown in Table 4.1, the agents can reproduce their own experiment or others' experiment.

4.2 Competency Questions

We have clearly defined and distinguished the terms "Reproducibility" and "Repeatability" in the previous section 4.1. In this section, we figure out the provenance information required for the reproducibility of a scientific experiment. We first understand the components required to describe a scientific experiment from the perspective of researchers. To do so, we conducted several oral interviews with scientists from different disciplines. As mentioned in Section 1.1, the requirements are driven from the scientists from Collaborative Research Center (CRC) ReceptorLight[1] where scientists work together to develop high-performance microscopy techniques. The different scientific practices followed in their experiments and their requirements of reproducibility and data management were gathered from interviews with the scientists in the CRC. In addition to that, a workshop to foster reproducible science[2] was conducted where scientists from Biology, Chemistry, Biodiversity, Ecology, and Computer Science participated. We collected the important things which they consider are required for reproducibility of scientific experiments in the form of competency questions. The competency questions from these oral interviews which were collected were from scientists from various projects performing different kinds of experiments. The relevance of these questions is further supported by their large overlap with competence questions obtained in other contexts, e.g. the provenance challenge [Moreau et al., 2008]. We selected the ones which were commonly told by scientists from these collected questions. We then generalized these questions to reflect what kind of provenance information of scientific experiments are required by the scientists. Here we present the most common competency questions of which answers are required to describe a scientific experiment [Samuel et al., 2018].

CQ1 What are the input and output variables of an experiment?

CQ2 Which are the methods and standard operating procedures used?

CQ3 Which are the files and materials that were used in a particular step?

[1] `http://www.receptorlight.uni-jena.de/`

[2] `http://fusion.cs.uni-jena.de/bexis2userdevconf2017/workshop/`

CQ4 Which are the steps involved in an experiment which used a particular material?

CQ5 Which are the instruments that are associated with an experiment and their settings when the output was generated?

CQ6 Which are the agents directly or indirectly responsible for an experiment?

CQ7 Who created this experiment and when? Who modified it and when?

CQ8 Which are the publications or external resources that were referenced in each step of an experiment?

CQ9 What is the complete path taken by a scientist for an experiment?

CQ10 List all the experiments which use growth protocol (EFO_0003789) and studies on "Homo sapiens" and resulted in phenotype "shorter prophase" which passed the quality control.

Question **CQ10** is an example query specific to life science experiments. In addition to the main competency questions **CQ1-CQ9**, specific questions related to each experiment are also asked by the scientists. For example, (1) What are the input variables of type Solution which were used in the bath solution preparation step of the experiments performed by an agent from a particular research group. (2) What are the output variables generated in the first execution of cell 4 of a particular Jupyter Notebook used as a Standard Operating Procedure in a particular experiment which used 'light sheet fluoroscence microscopy' method. The complete list of the competency questions is presented in Appendix B. To answer these type of questions, it is important to first model the provenance required to describe and reproduce the scientific experiments according to each discipline. Based on the competency questions, we studied the current provenance models (see Section 4.3) to understand whether they provide the elements required for describing the provenance of scientific experiments focusing on life-sciences.

4.3 Current Provenance Models

We already discussed briefly the provenance models in Chapter 3. In this section, we will see how some of these models have inspired our work and are being used in our ontology based on the competency questions described in Section 4.2. We will identify the aspects which are covered by these models and investigate how we use them and which extensions are required for describing provenance information based on our requirements.

Ram et al. present one such model which is called W7 model to represent the semantics of data provenance [Ram and Liu, 2006]. They identify the concepts to define the provenance in the context of events and actions. The W7 model presents seven different components of provenance and how they are related to each other.

Definition 4.3.1. Provenance is defined as a n-tuple P = (WHAT, WHEN, WHERE, HOW, WHO, WHICH, WHY, OCCURS_AT, HAPPENS_IN, LEADS_TO, BRINGS_ABOUT, IS_USED_IN, IS_BECAUSE_OF)
where P is the provenance; WHAT denotes the sequence of events that affect the data object; WHEN, the set of times of the event; WHERE, the set of locations of the event; HOW, the set of actions that lead to the events; WHO, the set of agents involved in the events; WHICH, the set of devices and WHY, the set of reasons for the event. OCCURS_AT is a collection of pairs (e, t) where e belongs to WHAT and t belongs to WHEN. HAPPENS_IN represents a collection of pairs (e, l) where l represents a location. LEADS_TO is a collection of pairs (e, h) where h denotes an action that leads to an event e. BRINGS_ABOUT is a collection of pairs (e, a_1, a_2,..a_n) where a_1, a_2,..a_n are agents who cooperate to bring about an event e. IS_USED_I is a collection of pairs (e, d_1, d_2,..d_n) where d_1, d_2,..d_n denotes devices. IS_BECAUSE_OF is a collection of pairs (e, y_1, y_2,..y_n) where y_1, y_2,..y_n denotes the reasons [Ram and Liu, 2006].

Another provenance model which also inspired our work is the PRIMAD [Freire et al., 2016] model. It describes a list of variables that could be changed or remain the same when trying to reproduce a study. They are as follows:

- **P** - Platform/Execution Environment/Context
- **R** - Research Objectives/Goals
- **I** - Implementation/Source Code/ Code
- **M** - Methods/Algorithms
- **A** - Actors/Persons
- **D** - Data (input data and parameter values)

The authors provide how a change in each variable of the PRIMAD model results in various types of reproducibility and the gain delivered to a computational experiment. For example, if only the Data (Parameters) are changed and rest is kept the same, then the reproducibility study tests the robustness of an experiment. If only the platform is changed and keeping the rest same, then the reproducibility study tests the portability of an experiment. When none of the variables in the PRIMAD data model are changed with the aim to verify whether the results are consistent, then the experiment is said to be repeated.
Another standard data model, PROV-DM, was introduced after the First, Second

and Third Provenance Challenges by the W3C working group [Belhajjame et al., 2013]. The PROV-DM is a generic data model to describe and interchange provenance between systems. It has a modular design with six components:

- *Entities and Activities*
- *Derivation of Entities*
- *Agents and Reponsibilities*
- *Bundles*
- *Properties that link entities*
- *Collections*

The PROV-O Ontology [Lebo et al., 2013] is the encoding of PROV-DM in OWL2 Web Ontology Language. It consists of a set of classes, properties and restrictions to implement provenance in different domains. The provenance information generated in different systems and contexts can be represented, exchanged and integrated using PROV-O. It is a generic vocabulary which can be directly used in different applications or can be extended further to meet the specific domain use cases. The authors encourage to extend this ontology to model provenance in fine or coarse granularity depending on the requirements of the users and applications. The terms in the ontology are grouped into three:

- *Starting Point* provide the basic elements of the PROV-O.
- *Expanded* consists of additional terms which are used to relate terms in the Starting Point.
- *Qualified classes and properties* provide additional attributes about the binary relations in the Starting Point and Expanded properties.

From this model, we have selected the terms which will be used in the development of the REPRODUCE-ME ontology (see Section 4.5) which are as follows:

- *prov:Entity* is a conceptual, physical or digital thing which can be either real or imaginary.
- *prov:Activity* is an event that happened over a period of time which resulted in processing, transformation or generation of entities.
- *prov:Agent* is something which is responsible for an activity. It has three subclasses:
  - *prov:Person* represents people.

  - *prov:Organization* represents a social institution.
  - *prov:SoftwareAgent* represents software.

- *prov:Location* represents a geographical place like Germany or a non-geographical place like a file.
- *prov:Plan* represents a set of actions.
- *prov:Collection* represents a collection of entities and provides a general structure to them.
- *prov:PrimarySource* represents the source which was generated without previous knowledge.
- *prov:Role* represents the function of an agent or an entity with respect to an activity.

From this model, we have selected the properties which will be used in the development of the REPRODUCE-ME ontology (see Section 4.5) which are as follows:

- *prov:startedAtTime* describes the time at which an activity started.
- *prov:endedAtTime* describes the time at which an activity ended.
- *prov:generatedAtTime* represents the time at which an entity is generated.
- *prov:invalidatedAtTime* represents the time at which an entity is invalidated or expired.
- *prov:wasInfluencedBy* relates an activity that influenced an entity.
  - *prov:actedOnBehalfOf* describes how an agent is responsible for an activity under the authority of another agent.
  - *prov:hadMember* represents the components of a Collection.
  - *prov:used* describes the usage of an entity by an activity.
  - *prov:wasAssociatedWith* describes how an activity is related to an agent.
  - *prov:wasGeneratedBy* relates a generation of a new entity by an activity.
  - *prov:wasAttributedTo* describes how an entity is attributed to an agent.
  - *prov:wasDerivedFrom* describes the derivation of a new entity from an existing one.
    - *prov:hadPrimarySource* represents the relationship between the derived entity out of the primary entity.
    - *prov:wasRevisionOf* represents the derived entity which is a revised version of the original entity.

  - *prov:wasInformedBy* represents the communication between two activities and how an entity generated by an activity is exchanged to other activity.
  - *prov:wasInvalidatedBy* represents the activity that invalidated the existence of an entity.
  - *prov:wasStartedBy* represents the agent who started an activity.
  - *prov:wasEndedBy* represents the agent who ended an activity.

- *prov:specializationOf* describes the relationship of two entities where one entity is a specialization of another.
- *prov:value* represents the value of an entity.
- *prov:invalidated* represents the activity that invalidated an entity.
- *prov:influenced* represents the ability of an activity, agent or an entity to make an influence on the characteristics of another.
- *prov:atLocation* relates an entity with its location.
- *prov:generated* relates an activity that generated an entity.

P-Plan [Garijo and Gil, 2012] is another model developed to describe the scientific workflows and their executions. The abstract scientific workflow is described as a plan which can be linked to the past executions. PROV provides *Plan* to describe the descriptions of scientific workflows, programs, and script. Since it is very broad and could not be able to describe further how the plans can be described and link to the past execution, P-Plan introduces the notion of *Steps* and *Variable*.
From this model, we have selected the class and property terms which will be used in the development of the REPRODUCE-ME ontology (see Section 4.5) which are as follows:

- *p-plan:Plan* is a subclass of *prov:Plan*. It consists of smaller steps which use and generate variables.
- *p-plan:Step* describes a planned execution activity.
- *p-plan:Variable* describes the input or output of the planned Activity.
- *p-plan:correspondsToStep* describes how an Activity is linked to its planned step.
- *p-plan:correspondsToVariable* describes how an entity which is associated with a planned activity is linked to a variable.

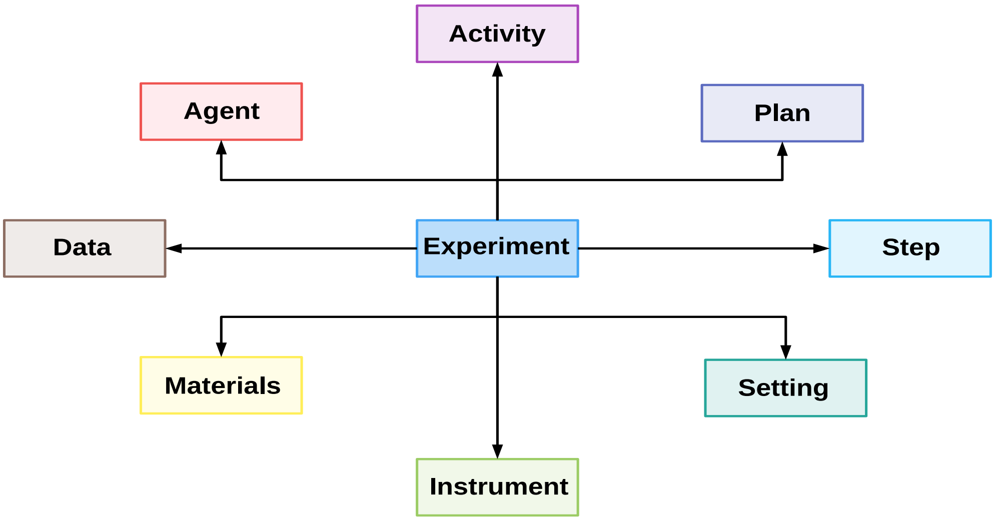


Figure 4.1: Overview of the REPRODUCE-ME data model to represent a scientific experiment

- *p-plan:hasInputVar* links the input variable to its planned step.
- *p-plan:hasOutputVar* links the output variable to its planned step.
- *p-plan:isInputVarOf* is the inverse relationship of *p-plan:hasInputVar* which links an input variable to its step.
- *p-plan:isOutputVarOf* is the inverse relationship of *p-plan:hasOutputVar* which links an output variable to its step.
- *p-plan:isPrecededBy* links a step to its preceding step.
- *p-plan:isStepOfPlan* links a step to its plan.
- *p-plan:isSubPlanOfPlan* links a plan to its bigger plan.
- *p-plan:isVariableOfPlan* links a variable to its plan.

4.4 The REPRODUCE-ME Data Model

Based on the problem statement (Chapter 2), we developed the REPRODUCE-ME (Reproduce Microscopy Experiments) data model [Samuel and König-Ries, 2017, Samuel, 2017]. It is a conceptual data model that forms a basis for the REPRODUCE-ME ontology. It is a generic data model for the representation of scientific experiments with their provenance information. The aim of this model is to capture the general elements of scientific experiments for their understandability and reproducibility. Figure 4.1 shows the overall view of the REPRODUCE-ME DM to represent a scientific experiment.

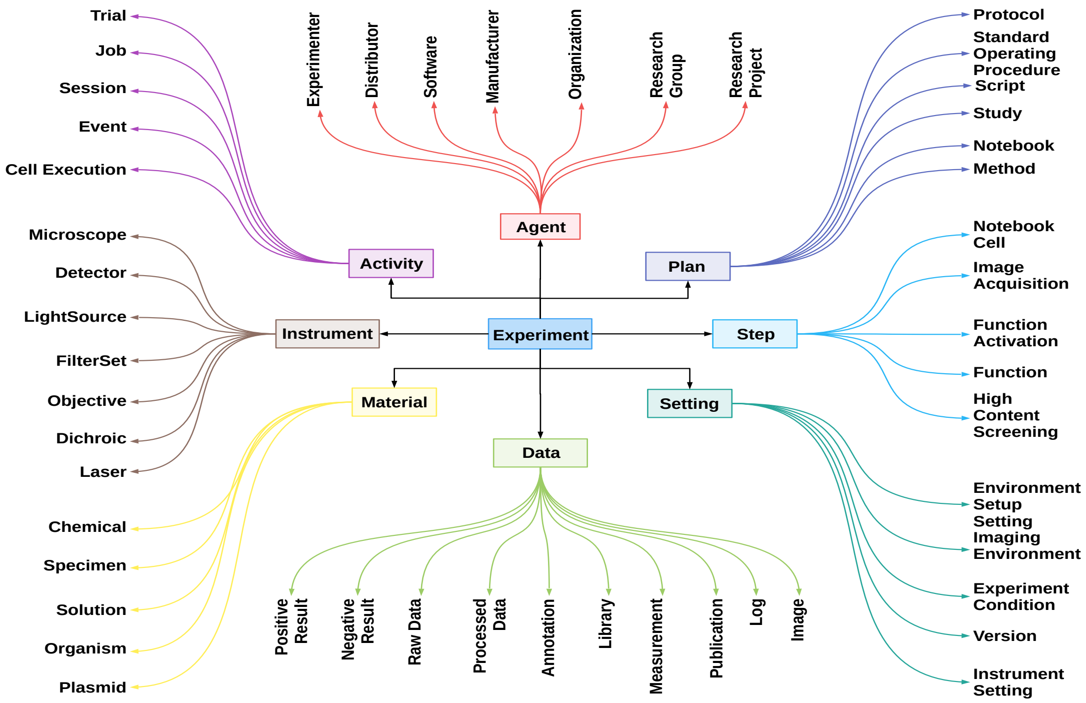


Figure 4.2: The expanded view of the REPRODUCE-ME data model used to represent a scientific experiment

An *Experiment* is considered as the central point of the REPRODUCE-ME data model. The model consists of eight components: Data, Agent, Activity, Plan, Step, Setting, Instrument, Material.

Definition 4.4.1. Experiment is defined as a n-tuple $E = (Data,\ Agent,\ Activity,\ Plan,\ Step,\ Setting,\ Instrument,\ Material)$

where E is the Experiment; *Data* denotes the set of data used and generated in E; *Agent*, the set of all people or organizations involved in E; *Activity*, the set of all activities occurred in E; *Plan*, the set of all plans involved in E; *Step*, the set of steps performed in E; *Setting*, the set of all settings; *Instrument*, the set of all devices used in E and *Material*, the set of all physical and digital materials used in E. The formal definition of each of these elements is given in the following sections. Figure 4.2 shows a part of the expanded view of REPRODUCE-ME data model for a scientific experiment.

Data

Definition 4.4.2. *Data* represents a set of data items used and generated in a scientific experiment E.

This is a fundamental part of a scientific experiment. The data in the PRIMAD model is a generic term which describes any data used in a study. The

REPRODUCE-ME data model further classifies the data. It is important to know which categories of data are important for reproducibility or repeatability. The data that is required to be shared to reproduce an experiment depends on each experiment. However, it is possible that there can be data which could belong to multiple subclasses. For example, a *Publication* from which a method or an algorithm is followed can either be an *Input Data* or it could be annotated as a *Final Result* of an experiment. Here, we categorize the data as follows:

- Metadata: Metadata is the data about data. It includes the *Temporal* and *Spatial* information, *Settings* and *Configurations*.
- Annotations: Annotations are notes that are added to the data in a text or multimedia files.
- Input Data: It is the data that is used as input to an experiment.
- Result: It is the data that is generated from an experiment. It could be further classified as follows:
    - Final Result: It is the final result that is generated in an experiment which is eventually used in a publication.
    - Intermediate Result: These results are obtained during the intermediate steps of an experiment.
    - Positive Result: These results which are annotated as positive are the results that confirm the hypothesis of an experiment.
    - Negative Result: These results are annotated as negative. This could be because of several reasons. The results which confute the hypothesis, the changes in experimental design or execution environment can cause negative results. However, the negative results can be important for other scientists because it would help them for better designing experiments.
- Parameters: These values are factors which define an operation or a system which is kept constant for a particular execution of an experiment or a calculation and varied over other executions.
- Raw Data: The data which has not been processed.
- Processed Data: The data which is processed after the generation of data.
- Measurements: The characteristic of an entity which is described as a numerical value which is used as a measure to compare with other entities.
- Publication: It is the textual description of an experiment including the research questions, hypothesis, methods, results, etc.

- Modified Version: It describes the version information of an entity which tells if there is any difference from its earlier form.
- License Document: It is the document which provides official permission to use or own an entity.
- Rights and Permissions Document: The document which tells whether other scientists are allowed to use or modify the data.

The *Data* can be seen as a subtype of *Entity* defined in the PROV data model.

Agent

Definition 4.4.3. *Agent* represent a group of people/organizations associated with one or many roles in a scientific experiment *E*.

Each agent is responsible for one or more roles in the activities and entities associated with an experiment. Some actors and their roles are extremely important for the understandability and reproducibility of a scientific experiment while others are less important or not applicable at all. For example, to know the name of a distributor of a sample/device is important in a biological study while it is less important or not applicable for a computer scientist. We present here the list of agents[3] who are directly or indirectly involved in a scientific experiment that were considered important based on our requirements:

- Experimenter: The person who performs an experiment.
- Manufacturer: The person who is responsible for the generation of an entity.
- Copyright holder: The person who holds the copyright or the permission of an entity or an activity.
- Distributor: The person who is responsible for the distribution of an entity or an experimental material.
- Author: The person who is the author of a publication.
- Principal Investigator: The person who supervises an experiment or a study.
- Contact Person: The person who acts as a corresponding person of a study to the scientific community.
- Owner: The person who owns an entity.
- Organization: A group of people.

[3] https://schema.org

  - Research Project: A group of people who are working together as a team in a project.
  - Research Group: A group of people who are working together as a team in a project.
  - Funding Agency: An organization responsible for granting funds for conducting a research project.

Activity

Definition 4.4.4. *Activity* represents a set of actions where each action has a starting and ending time which involves the usage or generation of entities in a scientific experiment E.

Activity is mapped to the *Activity* in PROV-DM model and extended with *Process* in PRIMAD model. It is a series of actions taken to achieve a task. The activities of scientific experiments are heavily dependent on their type and domain. Each trial of an experiment is considered as an activity. The executions of an experiment are important to understand how the final results are derived and generated. The paper [Ferro and Silvello, 2017] describes the system runs as the most important concern for reproducibility with regard to an Information Retrieval System. The output of the runs or the executions help other researchers to compare their new ideas with previous results of executions. The hidden parameters and settings in each execution can make a difference even if an experiment is performed with the same dataset and the same platform. Here we consider the attributes of activities of an experiment which are important.

- Execution Order: The order of execution is very important. For example, in a Jupyter Notebook, the cells can be executed in any order. The order of the execution will actually affect the result.

- Difference of executions: The difference in the source and the output of an execution of an experiment.

- Prospective Provenance: The provenance information of an activity that specifies its plan.

- Retrospective Provenance: The provenance information of what happened when an activity is performed.

- Causal Effects: The causal effects of an activity denotes the effects on an outcome because of another activity.

- Preconditions: The conditions that must be fulfilled before performing an activity.

- Cell Execution: The execution of a cell of a computational notebook is an example of an activity.
- Trial: The various tries of an activity. For example, several executions of script.

Plan

Definition 4.4.5. *Plan* represents a collection of steps and actions to achieve a goal.

The Plan is mapped to the *Plan* in the PROV-DM and P-Plan model. Here, we categorize the Plan as follows:

- Experiment: A scientific procedure with a coordinated set of steps and actions with the goal to test a hypothesis.
- Protocol: A specification with a set of instructions guiding how an activity is performed.
- Standard Operating Procedure: A set of step-by-step instructions to carry out a complex routine compiled and approved by an organization to use in specific environments.
- Method: A systematic procedure to accomplish a task.
- Algorithm: A set of rules or steps to be followed in a problem-solving operation.
- Study: A process to examine and analyze a data object to answer questions and discover new facts about it.
- Script: A computer program written to perform a task in a scripting language.
- Notebook: A computational notebook consists of a set of steps to perform computation and visualize the results inline.

Step

Definition 4.4.6. *Step* represents a collection of actions that represents the plan for an activity.

A Step represents a planned execution activity. The Step is mapped to the *Step* in the P-Plan model. Here, we categorize the Step as follows:

- Computational Step: The step or the process which uses computational resources involved in an experiment.

- Non-computational Step: The step or the process which does not depend on computational resources.
- Intermediate Step: The step which is performed during an experiment.
- Final Step: The step that is performed at the end of an experiment.

Setting

Definition 4.4.7. *Setting* represents a set of configurations and parameters involved in an experiment.

Here, we categorize the Settings as follows:

- Execution Environment: The execution environment of an experiment.
- Context: The background setting of an experiment.
- Instrument Settings: The settings and configuration of the devices that are used in an experiment.
- Computational Tools: The tools that use computer-based systems for computation.
- Packages: A collection of programs and resources which are packaged together.
- Libraries: A collection of resources used for the development of software.
- Software: The computer program which performs a particular task or tasks.

Instrument

Definition 4.4.8. *Instrument* represents a set of devices used in an experiment.

We model the scientific experiments by applying to high-end light imaging microscopy experiments. Hence, to include domain semantics, we add the terms which are related to microscopy. However, this element can further be extended based on the requirement of an experiment. Here, we categorize the Instruments as follows:

- Microscope: An instrument to observe and capture images.
- Detector: They are detectors which collect the photons emitted by the observed object which transforms the light signal into an electrical signal.
- LightSource: The source of light for a microscope.
- FilterSet: The set of filters which are either excitation or emission filters.

- Objective: The optical elements which are closest to the observed object which gathers light from the object and focuses the light to produce real images.
- Dichroic: A Dichroic Filter is an optical filter to selectively pass light of a small range of colors while passing other colors.
- Laser: The source for the laser light beam to focus light on the observed object.

Material

Definition 4.4.9. *Material* represents a set of physical or digital entities used in an experiment.

We model the scientific experiments in life sciences. Hence, we provide some of the materials related to life sciences which are added in the data model.

- Chemical: A pure substance with constant chemical composition and properties used in an experiment.
- Solution: A homogeneous mixture of substances. For example, a chemical solution used in an experiment.
- Specimen: Specimen is a part of a thing used in an Experiment or a study to determine the character of the whole thing.
- Plasmid: A small circular DNA strand usually found in the cytoplasm.

4.5 The REPRODUCE-ME Ontology for the Representation of Scientific Experiments

Based on the REPRODUCE-ME data model, we develop the ontology and extend with the components of a life science imaging experiment. The REPRODUCE-ME ontology [Samuel and König-Ries, 2017] was initially developed to represent the scientific experiments taking the real case scenario from life sciences. It is undergoing continuous development to model the scientific experiments in general irrespective of their domain. We first describe the methodological process that we followed in the development of the REPRODUCE-ME ontology. The development process is based on a collaborative approach [Holsapple and Joshi, 2002] using the guidelines for the ontology development [Noy et al., 2001].

The collaborative approach to ontology design proposes four phases in ontology engineering. In the *Preparation* phase, we define the design criteria, determine the boundary conditions and decide the evaluation standards. In the second phase, *Anchoring*, an initial ontology is produced to get the focus of the collaborators. We identified the collaborators and the critiques and comments are added to the ontology in the *Interactive Improvement* phase. The ontology is iteratively revised until a consensus is reached. In the last *Application* phase, the ontology is demonstrated and used within application. We now explain each phase in the development of the REPRODUCE-ME ontology.

- **Preparation**

  Based on the methodology that was described by [Holsapple and Joshi, 2002,Noy et al., 2001,Grüninger and Fox, 1995], we identified the requirements of the REPRODUCE-ME ontology. The domain of the ontology was first narrowed to the scientific experiments in the microscopy field. The major purpose of developing the ontology is to semantically represent the complete path of a scientific experiment including the computational and non-computational steps along with its execution environment. The scope of the ontology is to use it in the scientific data management platforms as well as the scripting tools that are used to perform computational experiments. We defined the implementation language required for the ontology to be OWL 2 since it is the most used language for developing ontologies. The REPRODUCE-ME ontology is available online along with the documentation[4].

  We identified the end-users of the ontology to be the domain scientists from life sciences who want to preserve and describe their experimental data in a structured format. The aim of the ontology is to represent the experimental data in an interoperable way that it can be used for understandability and reproducibility of results. The ontology could provide a meaningful link between the data, intermediate and final results, methods and execution environment which will help the scientists to follow the path used in the experiment.

  Based on the competency questions described in Section 4.2, we extracted a list of terms from the competency questions to represent the concepts and properties of the ontology. Based on these activities, we created an Ontology Requirement Specification Document (ORSD) which specifies the requirements that the ontology should fulfill [Suárez-Figueroa et al., 2009]. The ORSD is presented in Appendix B.

- **Conceptualization**

  In this phase, we work on the conceptualization of the scientific experiments. We identify the general concepts based on the list of competency questions. The general terms like *Experiment*, *Method*, *Step*, *Result* are extracted. Based on the extracted terms, we analyzed the existing ontologies to model the provenance of scientific experiments. The W3C recommendation, PROV-O, provides a generic model to capture provenance of different

[4] `https://w3id.org/reproduceme/`

systems. PROV-O provides the means to extend it based on the domain. Based on the competency questions, it can answer, we selected PROV-O as the upper-level ontology. In order to represent further the steps and the input and output of each step, we analyzed further and found P-Plan to suit our requirements. P-Plan also uses PROV-O as its upper ontology. The development of REPRODUCE-ME ontology was not done from scratch, but rather by reusing existing vocabularies.

A top-down approach is followed in the development of the ontology. The general concepts were taken from the existing vocabularies like PROV-O and P-Plan. PROV-O is a generic vocabulary while P-Plan is designed for representing scientific workflows. To represent scientific experiments, we added concepts which address models of experiments. Then the specialized classes were added to the most generic classes. Several properties were identified and categorized as object and data properties.

- **Implementation**

  The ontology is developed using the ontology tool editor, *Protege*[5]. The OWL 2 language is used for the development and RDF/XML is used for the serialization of the ontology. The naming convention used in the ontology is similar to the ones that are reused. PROV-O and P-Plan follow the CamelCase convention which is also followed in the REPRODUCE-ME ontology. The prefix used to denote the ontology is “repr”. The namespace of the ontology is “https://w3id.org/reproduceme#”.

- **Annotation**

  Several annotations have been added to the ontology to capture the provenance of the ontology. It includes the creator, when it was created and modified etc. It is important to track the different versions of the ontology.

- **Documentation and Publication**

  The ontology is documented using the WIDOCO tool [Garijo, 2017]. Using this tool, a set of HTML pages with diagrams and human-readable descriptions of the ontology terms are created. The ontology is documented and published online. The published ontology uses persistent URLs so that the ontology terms could be dereferenceable. The ontology can be downloaded in RDF/XML, TTL or N3 serializations. The ontology is publicly available[6].

- **Validation**

  The REPRODUCE-ME Ontology is validated using the OOPS tool[7]. It

[5] `https://protege.stanford.edu/`

[6] `https://w3id.org/reproduceme`

[7] `http://oops.linkeddata.es`

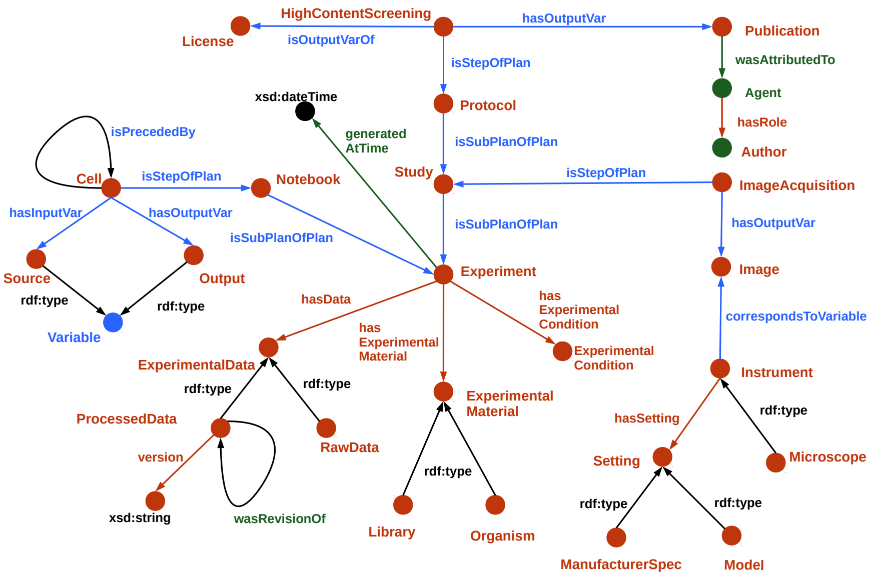


Figure 4.3: A scientific experiment depicted using the REPRODUCE-ME ontology [Samuel et al., 2018]

helped in detecting common pitfalls during its development. The pitfalls were corrected as and when they were found. The ontology is evaluated using it in application [Noy et al., 2001] which is mentioned in Chapter 7.

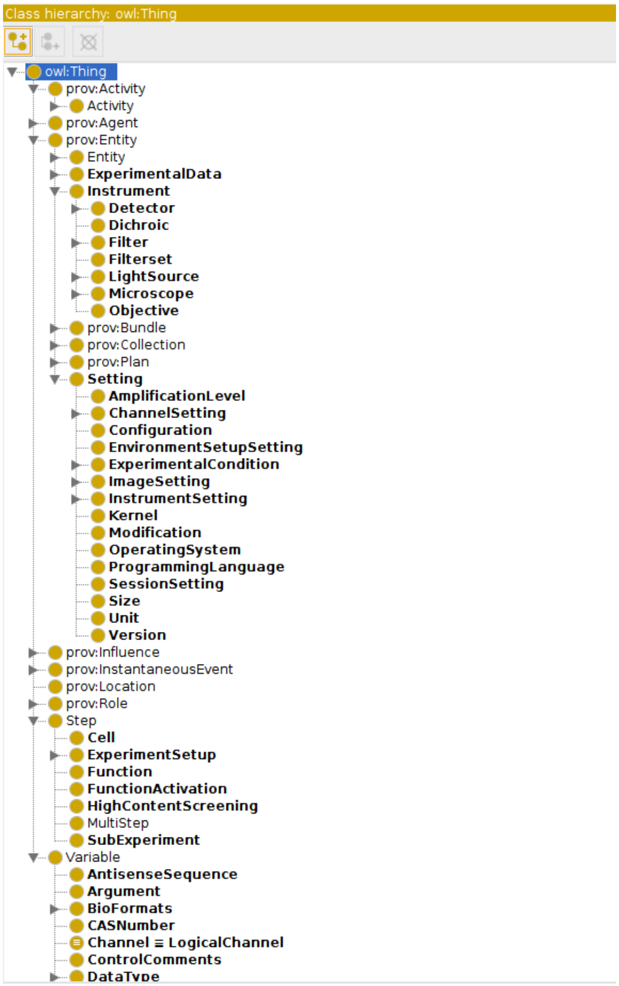
- **Versioning**
  To maintain the changes in the ontology, versioning of the ontology is maintained.

4.5.1 The REPRODUCE-ME Ontology

To describe the complete path of a scientific experiment, we encode the REPRODUCE-ME data model in OWL2 Web Ontology Language. The rationale behind doing so is to share a common understanding of the scientific experiment along with the domain knowledge among people and machines. Figure 4.3 shows an excerpt of the REPRODUCE-ME ontology terms depicting the lifecycle of a scientific experiment. Figure 4.4 shows an excerpt of REPRODUCE-ME ontology in Protege. The ontology is developed to represent the complete path of an experiment. The data elements that we identified in the REPRODUCE-ME data model are the base terms added in the ontology. To add these terms in the ontology, we first decide which class these concepts belong to in the upper ontologies PROV-O and P-Plan. In the development of our ontology, we add classes for every concept and reuse the properties from the upper ontologies PROV-O and P-Plan to describe the relationships between the added concepts.

CQ1 What are the input and output variables of an experiment?

The concept *Experiment* is added to represent the class of scientific experi-



Class hierarchy: owl:Thing

```
▼ owl:Thing
    ▼ prov:Activity
        ▶ Activity
    ▶ prov:Agent
    ▼ prov:Entity
        ▶ Entity
        ▶ ExperimentalData
        ▼ Instrument
            ▶ Detector
              Dichroic
            ▶ Filter
              Filterset
            ▶ LightSource
            ▶ Microscope
              Objective
        ▶ prov:Bundle
        ▶ prov:Collection
        ▶ prov:Plan
        ▼ Setting
              AmplificationLevel
            ▶ ChannelSetting
              Configuration
              EnvironmentSetupSetting
            ▶ ExperimentalCondition
            ▶ ImageSetting
            ▶ InstrumentSetting
              Kernel
              Modification
              OperatingSystem
              ProgrammingLanguage
              SessionSetting
              Size
              Unit
              Version
    ▶ prov:Influence
    ▶ prov:InstantaneousEvent
      prov:Location
    ▶ prov:Role
    ▼ Step
          Cell
        ▶ ExperimentSetup
          Function
          FunctionActivation
          HighContentScreening
          MultiStep
          SubExperiment
    ▼ Variable
          AntisenseSequence
          Argument
        ▶ BioFormats
          CASNumber
          Channel ≡ LogicalChannel
          ControlComments
        ▶ DataType
```

Figure 4.4: The REPRODUCE-ME ontology in Protege

ments conducted to test a hypothesis or perform a discovery. The *Experiment* is modeled as a *Plan* which consists of several steps and sub plans. The steps are related to the experiment using the object property *p-plan:isStepOfPlan* and the sub plan with the object property *p-plan:isSubPlanOfPlan.* The steps of an *Experiment* contains both input and output which are modelled as *p-plan:Variable.* The input and output variables are related to an experiment through steps using the properties *p-plan:isInputVarOf* and *p-plan:isOutputVarOf.* Each data element defined in *Data* (Section 4.4) is added as *p-plan:Variable.* Each step can either be computational or non-computational. Based on the domain, the input and output of scientific experiments can be added as variables. For example, *Image* is an output variable of the *Image Acquisition* step which is a major step in a life science experiment involving microscopy (see Figure 4.3). The *Protocol* is a sub plan of *Experiment* and it has many steps like *HighContentScreening.* In this way, we could describe the steps and plans and their input and output variables.

CQ2 Which are the methods and standard operating procedures used?
The concept *Method*, *Standard Operating Procedure* and *Protocol* are added to model the methods, standard operating procedures and protocols. These concepts are modeled as *Plan.* In addition to that, each element defined in *Plan* (Section 4.4) is also added as *p-plan:Plan.* These concepts are linked to the experiment using the property *p-plan:isSubPlanofPlan.* The object property *usedMethod* is used to show the relationship between a step of an experiment and the method.

CQ3 Which are the files and materials that were used in a particular step?
The concepts *ExperimentalMaterial* and *File* are added to model the general set of all experiment materials and files respectively. They are added as subclasses of a *prov:Entity* and *p-plan:Variable.* Since they are added as a *p-plan:Variable*, they can be linked to the steps of an *Experiment.* If the *ExperimentalMaterial* is an input to a step, it is linked to the step using the object property *p-plan:hasInputVar* or the inverse property *p-plan:isInputVarOf.*

CQ4 Which are the steps involved in an experiment which used a particular material?
The object property *p-plan:correspondsToVariable* relates an experiment to a variable. The variable is related to a step using object properties *p-plan:isInputVarOf* and *p-plan:isOutputVarOf.* Here we address the step which used a particular material, hence, we use the object property *p-plan:isInputVarOf.*

CQ5 Which are the instruments that are associated with an experiment and

their settings when the output was generated?
The instruments play an important role in the reproducibility of scientific experiments and it is important to describe them. We add *Instrument* as a *prov:Entity* to represent the set of all instruments or devices used in an experiment. The *Settings* are added as another class to represent the configurations in general. Each *Instrument* consists of several sub-parts which is represented using the object property *hasPart* and inverse property *isPartOf*. Each instrument as well the part of the instruments have settings which are described using the object property *hasSetting*.

CQ6 Which are the agents directly or indirectly responsible for an experiment?
We reuse the concepts of PROV-O to represent the agents responsible for an experiment. We add additional agents specialized for scientific experiments as mentioned in Section 4.4. We add terms like Author, ContactPerson, Distributor, Experimenter, Original Creator, Owner, Principal Investigator, Research Group, Research Project, Funding Agency, Distributor, Manufacturer to represent the agents who are directly or indirectly responsible for an experiment. We use the data property *ORCID*[8] to identify the agents of an experiment.

CQ7 Who created this experiment and when? Who modified it and when?
The temporal and spatial properties are important factors to know the provenance of scientific experiments. We reuse the object and data properties of PROV-O to answer these questions. We use the object property *prov:wasAttributedTo* to relate the experiment with the responsible agents. The properties *prov:generatedAtTime* and *modifiedAtTime* are used to describe the creation and modification time respectively.

CQ8 Which are the publications or external resources that were referenced in each step of an experiment?
Each step of an experiment has input or output variables. We add the concept *Publication* to the ontology. The publication used or generated in an experiment is described using the object properties *p-plan:isInputVarOf* and *p-plan:isOutputVarOf*. We use the properties *doi*[9], *pubmedid*[10], and *pmcid*[11] to identify the publications.

CQ9 What is the complete path taken by a scientist for an experiment?
We defined what is a scientific experiment and what are the essential elements needed to describe its provenance for reproducibility. To describe a complete

[8] https://orcid.org/

[9] https://www.doi.org/

[10] https://www.ncbi.nlm.nih.gov/pubmed/

[11] https://www.ncbi.nlm.nih.gov/pmc/

path of a scientific experiment, we need to describe the computational and non-computational steps and plans used in an experiment, the people who are involved in an experiment and their roles, the input and output data, the instruments used and their settings, the execution environment, the spatial and temporal properties of an experiment. To describe each item, we add the corresponding concepts to the ontology. We use the object property *p-plan:isPrecededBy* to represent the order of the steps performed to describe the complete path. In Chapter 5, we present the elements added in the ontology to describe the computational steps and plan. We also show how we interlink the computational steps to the main experiment.

4.6 Summary

This chapter presented precise definitions of reproducibility and repeatability of scientific experiments. Through oral interviews with scientists from different disciplines, we figured out the important provenance information required for reproducibility of scientific experiments in the form of competency questions. Based on that, we revisited the provenance models which inspired our work to see which aspects are already covered and what needs to be extended. This led to the development of the REPRODUCE-ME Data Model, which is one of the important contributions that we presented in this chapter. It is a conceptual data model to represent scientific experiments along with its provenance information developed taking into account also our requirements. We defined each variable of the data model in detail. We followed the methodology by [Holsapple and Joshi, 2002, Noy et al., 2001] in the development of the REPRODUCE-ME ontology. We maximized the reuse of existing standards. We focused on ten competency questions which include more general questions like **CQ1-CQ9** and more complex queries like **CQ10**. We provided details on how the terms in the REPRODUCE-ME ontology are used to answer these questions. We evaluate our work in detail in the upcoming chapters 6 and 7 by using it in applications mentioned by the ontology development guidelines [Noy et al., 2001]. We answer the competency questions with the help of SPARQL queries using the data provided by scientists working with the microscopy imaging techniques in Section 7.5.

Chapter 5

Computational Reproducibility

Computer programming which was once seen as a reserved skill for computer geeks has not only become a key prerequisite for researchers from different scientific fields but also plays a significant role in school education [Fessakis et al., 2013]. The use of computational tools has become vital in most of the scientific experiments to address the complexity and automation of tasks [Moreau and Tranchevent, 2012]. Scientists use scripting to perform computational tasks like data exploration and processing and link their input/output to their experiments. According to our survey described in Section 7.3 (see Figure 7.24), around 85% out of 101 participants write scripts to perform data analysis. Sharing scripts which are used for computational tasks is pretty straightforward and it is now becoming a mandatory prerequisite for publishing results in some accepted journals (see Chapter 1). One of the main requirements for computational reproducibility is also sharing of scripts/code.

Apart from scripts, computational notebooks have become one of the means to share code along with documentation [Shen, 2014]. There is a rapidly increasing use of computational notebooks among scientists, data analysts and even among teachers. These notebooks provide an interactive environment to write and run the code, and view graphical results inline. They allow users to perform data exploration, run simulations and visualize results by combining text and code together [Samuel and König-Ries, 2018b]. Interactive notebooks are not just used for performing computational tasks but also for documenting and sharing their results. The primary objective of computational notebooks as described by the Project Jupyter team [Kluyver et al., 2016] is to provide *the collaborative creation of reproducible computational narratives that can be used across a wide range of audiences and contexts*[1]. One of the major reasons for their widespread adoption among scientists is because they enable computational reproducibility. Even though these notebooks are meant for reproducible science, the provenance

[1] https://blog.jupyter.org/project-jupyter-computational-narratives-as-the-engine-of-collaborative-data-science-2b5fb94c3c58, Blog: Accessed on January 29, 2019

support in them is limited [Rule et al., 2018, Samuel and König-Ries, 2018b]. Small changes and errors during the data collection, processing and documenting phases can lead to significant changes in results [Samuel et al., 2018]. Tracking which code/function resulted in a particular output becomes cumbersome with limited provenance support. Hence, the presence of provenance support in these notebooks for computational reproducibility is desirable.

As discussed in Section 3.1.2, there are some tools which support computational reproducibility by capturing provenance from scientific workflows or scripts. However, there are only a few provenance capturing techniques for the execution of Jupyter notebooks. One such approach is the integration of noWorkflow in IPython notebooks [Pimentel et al., 2015]. The provenance collected from an external script using noWorkflow can be analyzed and displayed in the notebook using the line magic "%now_run". The visualization of the provenance information of the external script is shown inside the IPython notebooks and thus scientists do not need to switch the environments. Through this approach, only the provenance of external scripts can be analyzed and visualized. But it does not solve the problem of tracking the provenance of the execution of the code inside the Jupyter Notebook. Another limitation of this approach is that it only works for python code in Jupyter notebooks. Therefore, it could not cope with the vast number of programming languages[2] supported by these notebooks. The lack of semantic representation of scripts is another research problem that needs to be addressed.

In this chapter, we envision a novel solution for supporting computational reproducibility through Jupyter notebooks. We present ProvBook, an extension of Jupyter notebooks to capture its provenance information [Samuel and König-Ries, 2018b, Samuel and König-Ries, 2018c]. This framework provides an easy-to-use environment for the scientists and developers for the efficient visualization of the provenance data. Based on the functional and non-functional requirements described in Section 2.5, we show in Figure 5.1 the key components provided by ProvBook to help in the end-to-end management of provenance for computational reproducibility. We show how these components which are vital in supporting computational reproducibility are developed and used in ProvBook.

This chapter provides the background, structure, and workflow of Computational Notebooks in Section 5.1. We introduce ProvBook in Section 5.2. How ProvBook provides support for the provenance capture and management in Jupyter Notebooks is described in Section 5.2.1. The semantic representation of computational notebooks and their execution using the REPRODUCE-ME ontology is presented in Section 5.2.2. The provenance difference of several executions of a Jupyter Notebook is presented in Section 5.2.3. Section 5.3 presents how scripts and their execution are described using the REPRODUCE-ME ontology. This chapter

[2] `https://jupyter4edu.github.io/jupyter-edu-book/` Blog: Accessed on April 11, 2019

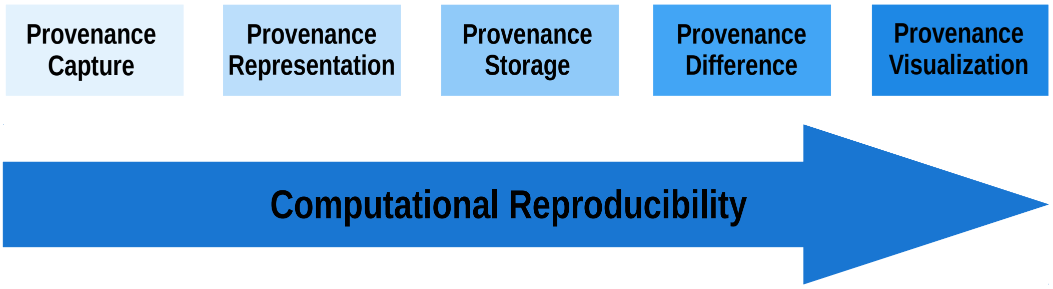


Figure 5.1: The key components for the end-to-end provenance management provided by ProvBook for computational reproducibility

concludes with the summary (Sections 5.4). Parts of the results of this chapter have been published in [Samuel and König-Ries, 2018a, Samuel and König-Ries, 2018b, Samuel et al., 2018].

5.1 Computational Notebooks

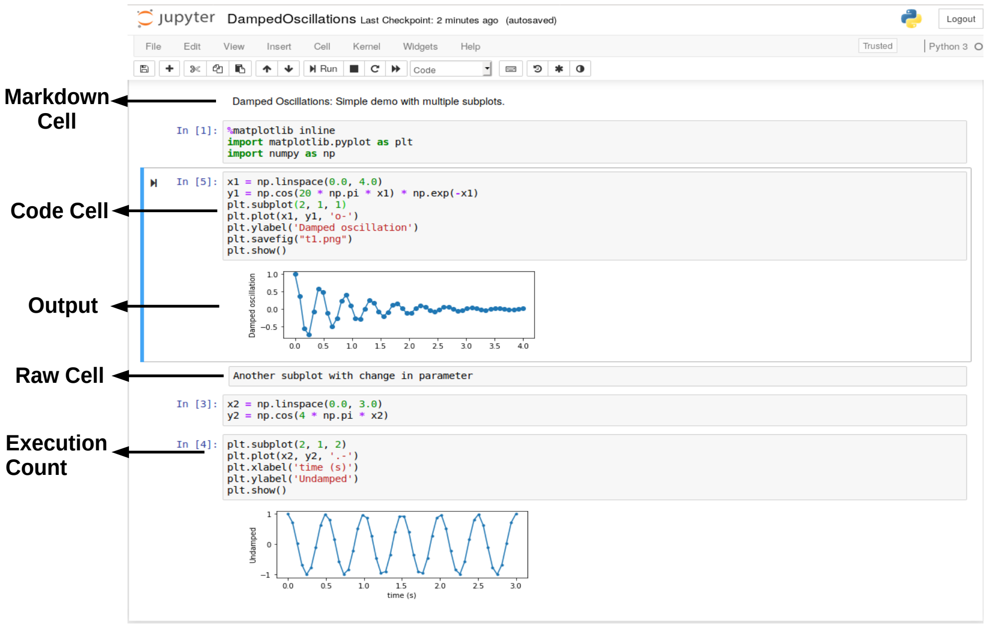
A computational notebook is a virtual environment for literate programming. The term "Literate Programming" was first introduced by Donald Knuth [Knuth, 1984]. It is a programming paradigm where the source code is combined with the explanation of the program logic using natural language. The computational notebooks were first available in 1988 with the release of the proprietary software, Mathematica [Wolfram, 1988]. It was then followed by another proprietary software, Maple[3], which released its notebook-style graphical user interface in 1989. The computational notebooks have gained widespread adoption in the past decade due to the emergence of free and open source platforms like Project Jupyter [Kluyver et al., 2016] and RStudio [Team et al., 2015]. According to Project Jupyter, there are over 1.7 million Jupyter notebooks publicly hosted on Github and have millions of users from several disciplines[4].

5.1.1 Background and Structure of Jupyter Notebooks

A Jupyter Notebook, formerly known as IPython Notebook, is a web-based application which provides an interactive and computational environment [Kluyver et al., 2016]. Figure 5.2 shows a sample Jupyter Notebook. It is organized as a sequence of ordered cells. A cell is a multiline text input field. There are three types of cell:

[3] `https://www.maplesoft.com/products/Maple/`

[4] `https://blog.jupyter.org/jupyterlab-is-ready-for-users-5a6f039b8906`, Blog: Accessed on January 29, 2019



DampedOscillations Last Checkpoint: 2 minutes ago (autosaved)

Damped Oscillations: Simple demo with multiple subplots.

```
In [1]: %matplotlib inline
        import matplotlib.pyplot as plt
        import numpy as np
```

```
In [5]: x1 = np.linspace(0.0, 4.0)
        y1 = np.cos(20 * np.pi * x1) * np.exp(-x1)
        plt.subplot(2, 1, 1)
        plt.plot(x1, y1, 'o-')
        plt.ylabel('Damped oscillation')
        plt.savefig("t1.png")
        plt.show()
```

```
Another subplot with change in parameter
```

```
In [3]: x2 = np.linspace(0.0, 3.0)
        y2 = np.cos(4 * np.pi * x2)
```

```
In [4]: plt.subplot(2, 1, 2)
        plt.plot(x2, y2, '.-')
        plt.xlabel('time (s)')
        plt.ylabel('Undamped')
        plt.show()
```

Figure 5.2: A sample Jupyter Notebook

- Code cell

  A code cell allows to edit and run code based on the kernel. The default kernel is Python which runs Python code. The code cell can be executed using the "Play" button in the toolbar or the "Run Cell" from the Cell menu bar in the notebook interface. The code is sent to the associated kernel when it is executed and the result of the output is displayed in the notebook under the executed cell as shown in Figure 5.2.

- Markdown cell

  The Notebook allows documenting the computational process in a literate programming way, where the users can explain the code using text. This is possible by marking up the text with the Markdown language in the cells which are called Markdown cells. The parts of the text can be made italics, bold, etc. using the markup language. It also provides the facility to include inline mathematics using the Latex notation: $...$ and $$...$$ for displayed mathematics. The markdown text is converted into the corresponding formatted rich text on the execution of a markdown cell.

- Raw cell

  A raw cell allows writing the output directly. The raw cells are not evaluated or formatted by the notebook.

The notebook cells can be grouped and reordered by simply clicking on the cell and dragging them to a particular place. The cells can be executed either linearly or in any order. The ability to execute the cells in any order distinguishes the computational notebooks from the traditional scripting tools. The cells are executed one at a time. When a cell is created, the notebook assigns it an identifier in incrementing order. The cell is re-assigned with an identifier every time the cell is re-executed. As we see in Figure 5.2, the highlighted code cell in the third position has an execution count of 5, while the other code cells after it (in the order of position) have a lesser execution count. This shows that computational notebooks can be executed in any order.

These notebooks also have rich display capabilities because the output of a code cell can be displayed in many rich representations [Kluyver et al., 2016]. They are: (1) HTML, (2) JSON, (3) PNG, (4) JPEG, (5) SVG, (6) LaTeX. Audio and Video files can be also embedded and rendered in a cell.

Different computational kernels are supported by Jupyter Notebook including Python, R, MATLAB, and Julia. The notebooks are saved with the *.pynb* extension and they are internally stored in JSON format. It can be installed locally in a user's computer or remotely. Currently, notebooks can be exported in different formats like HTML, LaTeX, PDF, Markdown from the command line or the user interface. This is possible using the nbcovert[5] tool provided by Jupyter Notebook. In this way, notebooks can be shared in multiple ways. The Jupyter Notebook Viewer (nbviewer) is used to view a notebook from a URL and rendered as a static web page. Using nbviewer, a notebook can be viewed remotely without installing it locally.

5.1.2 Workflow of Jupyter Notebooks

Scientists working on computational experiments divide and organize their tasks into cells. The cells are edited multiple times until they reach their expected results. They are executed one at a time, allowing the users to make multiple edits and re-execute them in any order. This is unlike the traditional scripts where the entire script is executed all at once. Users can do many modifications to the notebooks like inserting, removing or rearranging the cells. These operations could derive new results which could be different from that of the past executions. One of the ten simple rules for computational reproducible research as discussed in the paper [Sandve et al., 2013] is to record all intermediate results in a standardized format. Currently, only the output from the latest execution is stored in the notebook. The provenance of the final and intermediate results are not supported in the Jupyter notebooks. There is also no support to compare the intermediate results of the different executions. A recent study from 2018 [Rule et al., 2018] analyzed over 1 million publicly available notebooks from GitHub and interviewed 15 data scientists from different disciplines. One of the highlights based on their results is the need for tracking provenance especially when the cells are over-written and re-run. The provenance

[5] https://github.com/jupyter/nbconvert

information is substantially helpful especially in the machine learning experiments where it is essential to track how exactly a final result has been achieved. It is also necessary to keep track of the experiments that have been attempted because that may benefit other scientists, even if the results are negative or not as expected. The need for the provenance support in these computational notebooks resulted in the development of ProvBook.

5.2 ProvBook: Provenance of the Notebook

ProvBook[6] [Samuel and König-Ries, 2018b, Samuel and König-Ries, 2018c] is developed as an extension of Jupyter Notebook which supports capture and management of provenance information of its different executions over the course of time. The ProvBook provides four main features supporting the end-to-end provenance management of Jupyter Notebook for computational reproducibility as shown in Figure 5.1. They are:

1. Provenance Capture and Management
2. Semantic Representation
3. Provenance Difference
4. Provenance Visualization

Figure 5.3 shows the architecture of ProvBook which shows how a user can use it for computational experiments.

5.2.1 Provenance Capture and Management

One of the key challenges in developing provenance capture systems is to decide at what level of granularity the provenance needs to be collected. The motivation behind the development of ProvBook is to help scientists who use computational notebooks for their data analysis and exploration. In order to help users from every discipline irrespective of their programming skills, the design of ProvBook is kept simple so that it could be adopted as an easy-to-use tool. The Provenance Capture and Management modules of ProvBook are responsible to capture and store the provenance of the execution of the cells over the course of time. Every time the code cell is executed, the provenance of the execution is stored in the metadata of the cell. The provenance information of the cell execution includes the start and end time of each execution, the total time it took to run the code cell, the source code and the output got during that particular execution.

[6] `https://github.com/Sheeba-Samuel/ProvBook`

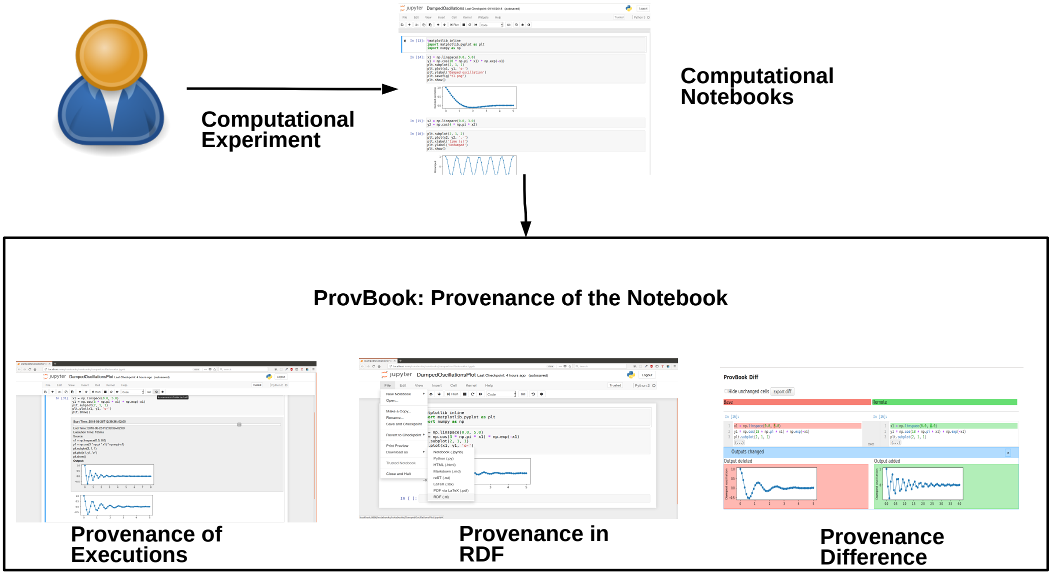


Figure 5.3: The architecture of ProvBook

5.2.1.1 The format of the Provenance of the Notebook

In order to understand how the provenance information is captured and stored, we first explain the detailed structure of Jupyter Notebooks. A Jupyter Notebook is stored as a JSON file format. Figure 5.4 shows a sample Jupyter Notebook metadata[7]. It is a dictionary with the following keys:

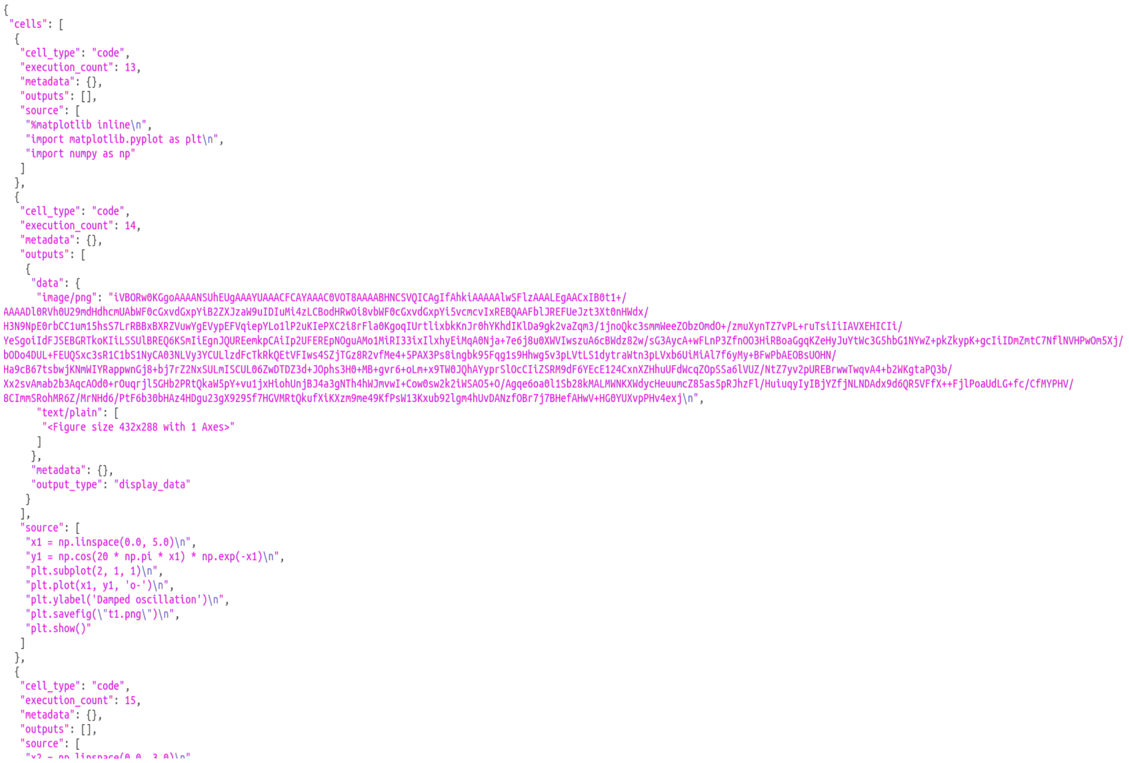
- metadata
- nbformat
- nbformat_minor
- cells

Metadata

Metadata is a dictionary that contains information about the notebook, its cells, and outputs. The notebook metadata defines the following keys:

- kernelspec
  This defines the kernel specifications of the notebook. The metadata includes the name of the kernel as displayed in the user interface, the name, and the display name of the language of the kernel.
- language_info
  This defines the name and version of the programming language of the kernel.

[7] `https://nbformat.readthedocs.io/`, Blog: Accessed on January 29, 2019



```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 13,
   "metadata": {},
   "outputs": [],
   "source": [
    "%matplotlib inline\n",
    "import matplotlib.pyplot as plt\n",
    "import numpy as np"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 14,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "image/png": "iVBORw0KGgoAAAANSUhEUgAAAYUAAACFCAYAAAC0VOT8AAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAALEgAACxIB0t1+/
AAAADl0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uIDIuMi4zLCBodHRwOi8vbWF0cGxvdGxpYi5vcmcvIxREBQAAFblJREFUeJzt3Xt0nHWdx/
H3N9NpE0rbCC1um15hsS7LrRBBxBXRZVuwYgEVypEFVqiepYLo1lP2uKIePXC2i8rFla0KgoqIUrtlixbkKnJr0hYKhdIKlDa9gk2vaZqm3/1jnoQkc3smmWeeZObzOmdO+/zmuXynTZ7vPL+ruTsiIiIAVXEHICIi/
YeSgoiIdFJSEBGRTkoKIiLSSUlBREQ6KSmIiEgnJQUREemkpCAiIp2UFEREpNOguAMo1MiRI33ixIlxhyEiMqA0Nja+7e6j8u0XWVIwszuA6cBWdz82w/sG3AycA+wFLnP3ZfnOO3HiRBoaGgqKZeHyJuYtWc3G5hbG1NcwZ8pkZpw0N/WV0S2y6wdSFDfNsYv4cn2Ih1KzzU/pmS08tPd2uIF8HYDOZ3jbi+k9ptqz/8rXmDmBjUnmjNAlN6ZYGbf6+cGbNj/oqz5+5gONC9fNaI4+Cw7k+2NTnN+eU1jGAb0o0rzRxgB2uVaMRqgXHvZL6XBrZf/3K9u9zeWP7uf6ZmFT+S98S/aFSlJTvqmJ7pj0qxSNPhgNjoc49MxHHNnw6Rbq1zyeO99MWWfOycRB9U0u1MxHuJmYzLhcbwU0fJu3KjU1pxmKfzYpDvgpbfJqmMZxuO/6+PpanXOP9QJjTSfI7Gu0m+ryY8yGhHbHHSDxxXU0aaKOLbB+eZMzVXlC49MRhL85eUu6yPKrlMSwLNnq3O/G0WOvTi+V9z2rGKi2Nuu1/2hdH5Rh+a80BIzbfrv53aZ3LBc9LfCJ6fgEHpUzdA==
  },
  "source" : "[some multi-line code]",
  "outputs": [{
      # list of output dicts
      "output_type": "stream",
      ...
  }],
}
```

Listing 5.2: Metadata structure for code cells

The execution count is the cell identifier which tells the count of the execution of the cell. The "outputs" of a code cell is the list of outputs got when the cell is executed. There are four types of output.

1. **stream**

```
{
  "output_type" : "stream",
  "name" : "stdout", # or stderr
  "text" : "[multiline stream text]",
}
```

Listing 5.3: Metadata structure for 'stream' output

2. **display_data**

```
{
  "output_type" : "display_data",
  "data" : {
    "text/plain" : "[multiline text data]",
    "image/png": "[base64-encoded-multiline-png-data]",
    "application/json": {
      # JSON data is included as-is
      "json": "data",
    },
  },
  "metadata" : {
    "image/png": {
      "width": 640,
      "height": 480,
    },
  },
}
```

Listing 5.4: Metadata structure for ‘display_data’ output

3. **execute_result**

```
{
  "output_type" : "execute_result",
  "execution_count": 42,
  "data" : {
    "text/plain" : "[multiline text data]",
    "image/png": "[base64-encoded-multiline-png-data]",
    "application/json": {
      # JSON data is included as-is
      "json": "data",
    },
  },
  "metadata" : {
    "image/png": {
      "width": 640,
      "height": 480,
    },
  },
}
```

Listing 5.5: Metadata structure for ‘execute_result’ output

4. **error**

```
{
  'output_type': 'error',
  'ename' : str,   # Exception name, as a string
  'evalue' : str,  # Exception value, as a string

  # The traceback will contain a list of frames,
  # represented each as a string.
  'traceback' : list,
}
```

Listing 5.6: Metadata structure for ‘error’ output

More metadata keys can be added at the cell level. They are:

- name: A string which describes the name of the cell
- tags: A list of strings Tags added to the cell
- collapsed: A bool value to check whether the container of the output of a cell is collapsed or not
- scrollable: A bool value to check whether the output of a cell is scrollable or not
- deletable: A bool value to check whether the cell is deletable or not
- format: The mime-type of a raw cell
- source_hidden: A bool value to describe whether the source of the cell is shown or hidden
- output_hidden: A bool value to describe whether the output of the cell is shown or hidden

The provenance information captured by the ProvBook is added to the content of the Notebook in the JSON format. The Notebook allows adding custom metadata to its content. So we add a list of dictionary of provenance metadata. The structure of the provenance in the metadata of the notebook is as follows:

```
"metadata": {
    "provenance": [
     {
      "end_time": "2019-02-08T11:21:42.352Z",
      "execution_time": "8ms",
      "outputs": [
       {
        "name": "stdout",
        "output_type": "stream",
        "text": "(’mean of data: ’, 3.9815812547037313)\n"
       }
      ],
      "source": "data = rnd.normal(loc=4, scale=2, size=100)\nprint('mean of data: ', np.mean(data))",
      "start_time": "2019-02-08T11:21:42.344Z"
     },
     {
      "end_time": "2019-02-08T11:21:54.818Z",
      "execution_time": "7ms",
      "outputs": [
       {
        "name": "stdout",
        "output_type": "stream",
        "text": "('mean of data: ', 3.969583207007243)\n"
       }
      ],
      "source": "data = rnd.normal(loc=4, scale=2, size=300)\nprint('mean of data: ', np.mean(data))",
      "start_time": "2019-02-08T11:21:54.811Z"
     }
    ],
}
```

Listing 5.7: Metadata structure for provenance added by ProvBook

The metadata added to the provenance by ProvBook are:

- **start_time**: The time at which the execution of the cell started
- **end_time**: The time at which the execution of the cell ended
- **source**: The input of the cell at a particular execution
- **outputs**: The outputs of the cell at a particular execution
- **execution_time**: The total time it took for the execution of a cell

The time of execution for a computational task in a Notebook is important to check the performance of the task. Therefore, the execution time was added as part of the provenance metadata. The start and end time also act as an indicator of the execution order of the cells. It is important for the user to check when a particular cell was last executed because the cells can be executed in any order as we have seen in Section 5.1.1. The users of computational notebooks make changes to the parameters and run a set of cells several times till they arrive at their expected result. It is important to track the history of all the executions to see what parameters were changed and how the results were derived. In order to do so, the input and output of each execution of a cell are saved in the notebook's content.

5.2.2 Semantic Representation

Many tools have emerged to capture the prospective and retrospective provenance of the scripts. There are represented and stored in different ways in different systems. Most of them store the provenance data in a traditional database in the provenance capturing systems of scripts. In our work, we aim to semantically represent the provenance information of script execution. This will help to describe the whole experiment semantically including the computational provenance as well. The provenance information of the scripts will be combined with other experimental metadata thus providing the context of the results. Thus we aim to make the experiments understandable along with their context. In this section, we focus on how we convert the computational notebooks into Resource Description Framework (RDF). While, in Section 5.3, we discuss how we convert the scripts in general into RDF.

ProvBook provides the user the ability to convert the notebooks into RDF along with the provenance traces and execution environment attributes. The REPRODUCE-ME ontology is used to describe the computational tasks of the notebook. The ontology is extended from PROV-O and P-Plan to describe the provenance information of the notebook.

We define the competency questions required to answer the questions related to the computational provenance.

CQ11 What is the complete path taken by a user for a computational notebook experiment?

CQ12 What is the sequence of steps in the execution of a computational notebook?

CQ13 How many trials were performed for a particular cell in a computational notebook?

CQ14 How long it took for a particular trial of a computational notebook?

CQ15 What was the source for a particular trial of a computational notebook?

CQ16 What was the output for a particular trial of a computational notebook?

CQ17 Who are the agents responsible for a computational notebook?

CQ18 When was a particular trial of a computational notebook last executed?

CQ19 What are the environmental attributes of a notebook execution?

The aim of this module is to semantically describe the prospective and retrospective provenance of a computational notebook. The module contains the concepts needed to represent the different elements of a computational notebook and the

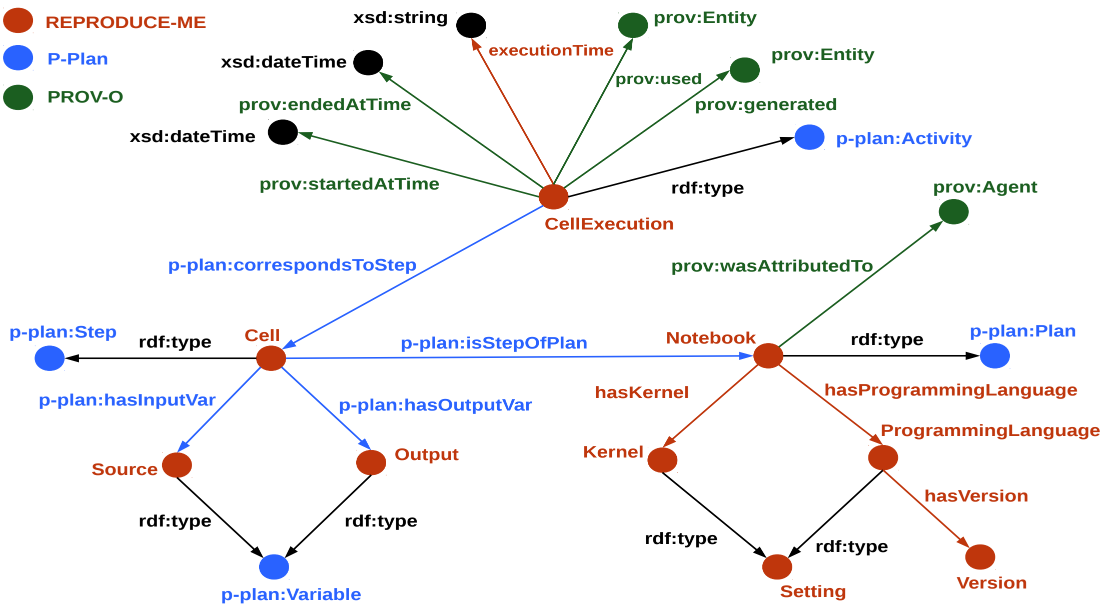


Figure 5.5: The semantic representation of a computational notebook [Samuel and König-Ries, 2018b]

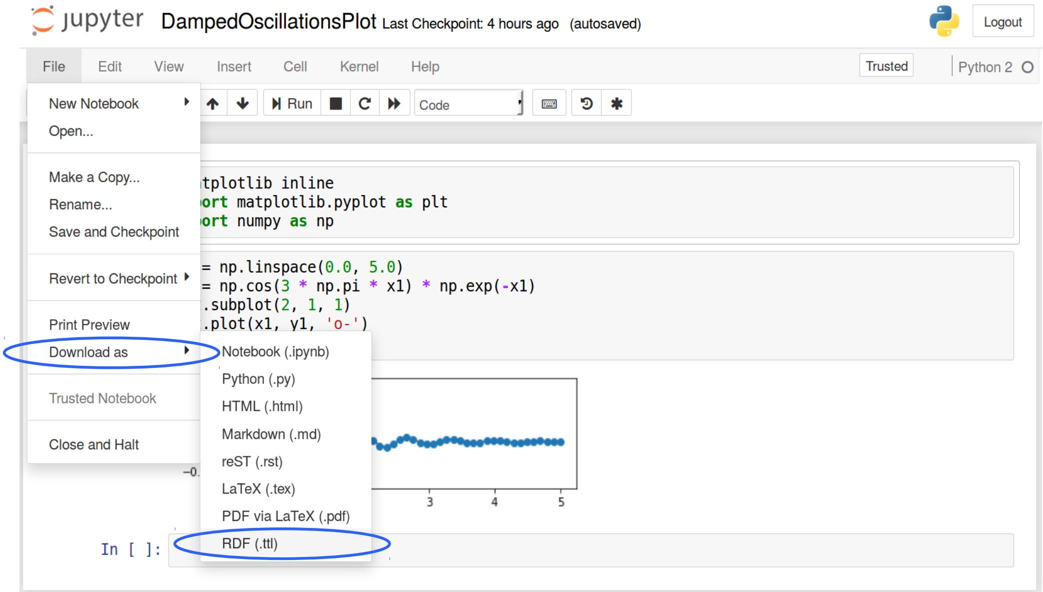
properties to relate the several trials of the notebook. We use RDF to represent the computational notebooks as we have discussed the benefits of using semantic web technologies to represent provenance information in Chapter 4. Figure 5.5 shows the semantic representation of a computational notebook. We define how the notebook is semantically described.

- Notebook
  The computational notebook is represented as a *Notebook* which is a subclass of *p-plan:Plan*. The *Setting*s describes the execution environment of the *Notebook*. The *Setting*s are *Kernel*, *ProgrammingLanguage*, *Version*.

- Cell
  The cell of a notebook is represented as *Cell* which is a *p-plan:Step*. The *Cell* is a step of *Notebook* and the relationship is described using *p-plan:isStepOfPlan*.

- Source
  The input of each cell is described as *Source* which is related to *Cell* using the object property *p-plan:hasInputVar*. The *Source* is a *p-plan:Variable*. The value of the *Source* variable is represented using *rdf:value*.

- Output
  The output of each cell is described as *Output* which is related to *Cell* using the object property *p-plan:hasOutputVar*. The *Output* is a *p-plan:Variable*. The value of the *Output* variable is represented using *rdf:value*.



```
tplotlib inline
ort matplotlib.pyplot as plt
ort numpy as np
```

```
= np.linspace(0.0, 5.0)
= np.cos(3 * np.pi * x1) * np.exp(-x1)
.subplot(2, 1, 1)
.plot(x1, y1, 'o-')
```

```
In [ ]:
```

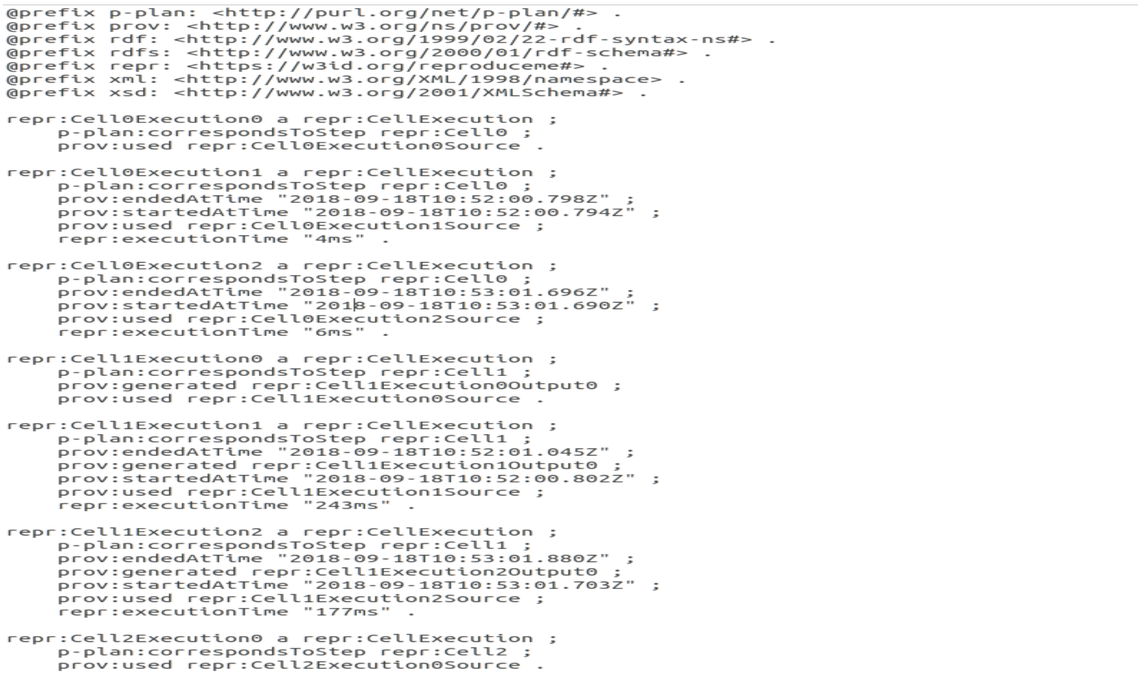
Figure 5.6: A Notebook which can be downloaded in RDF

- CellExecution
  Each execution of a cell is described as *CellExecution* which is a *p-plan:Activity*. The input of each *Execution* is an *prov:Entity* which is related using the property *prov:used*. The output of each *Execution* is an *prov:Entity* which is related using the property *prov:generated*. The data properties *prov:startedAtTime*, *prov:endedAtTime* and *repr:executionTime* are used to represent the starting time, ending time and the total time taken for the cell execution respectively.

Figure 5.6 shows a notebook which can be downloaded in RDF using ProvBook. The RDF can be downloaded as a turtle file either from the user interface of the notebook or using the command line. Figure 5.7 shows a part of Jupyter Notebook in RDF represented using REPRODUCE-ME ontology. It allows the user to share a notebook along with its provenance in RDF and also convert it back to a notebook. ProvBook also provides a reproducibility service where the provenance graph is converted back to a computational notebook along with its provenance. The provenance graph of the notebook can be converted back to a notebook using the command line. We answer the competency questions (**CQ11-CQ19**) using SPARQL in Chapter 7.

5.2.3 Provenance Difference

Reproducibility is the ability of a third party to reproduce the results generated from the description of its input and steps with the aim to confirm the original experimenter's results. Therefore, it is important to get the provenance information to reproduce the original experimenter's results. The provenance information includes the data and the steps along with the original results. In this module of ProvBook,

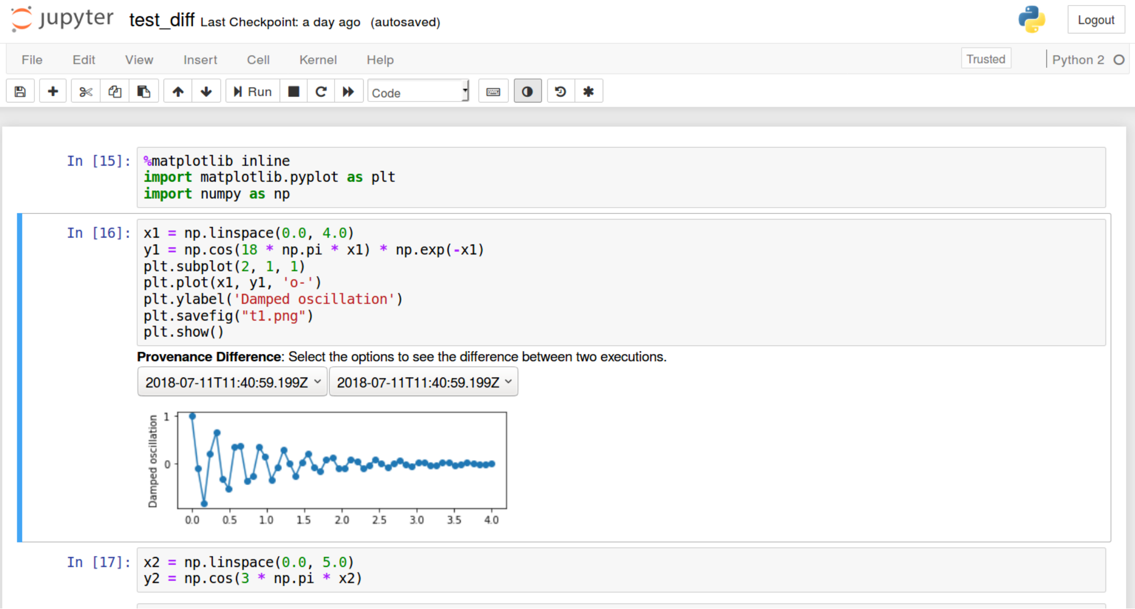


```
@prefix p-plan: <http://purl.org/net/p-plan/#> .
@prefix prov: <http://www.w3.org/ns/prov/#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix repr: <https://w3id.org/reproduceme#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

repr:Cell0Execution0 a repr:CellExecution ;
    p-plan:correspondsToStep repr:Cell0 ;
    prov:used repr:Cell0Execution0Source .

repr:Cell0Execution1 a repr:CellExecution ;
    p-plan:correspondsToStep repr:Cell0 ;
    prov:endedAtTime "2018-09-18T10:52:00.798Z" ;
    prov:startedAtTime "2018-09-18T10:52:00.794Z" ;
    prov:used repr:Cell0Execution1Source ;
    repr:executionTime "4ms" .

repr:Cell0Execution2 a repr:CellExecution ;
    p-plan:correspondsToStep repr:Cell0 ;
    prov:endedAtTime "2018-09-18T10:53:01.696Z" ;
    prov:startedAtTime "2018-09-18T10:53:01.690Z" ;
    prov:used repr:Cell0Execution2Source ;
    repr:executionTime "6ms" .

repr:Cell1Execution0 a repr:CellExecution ;
    p-plan:correspondsToStep repr:Cell1 ;
    prov:generated repr:Cell1Execution0Output0 ;
    prov:used repr:Cell1Execution0Source .

repr:Cell1Execution1 a repr:CellExecution ;
    p-plan:correspondsToStep repr:Cell1 ;
    prov:endedAtTime "2018-09-18T10:52:01.045Z" ;
    prov:generated repr:Cell1Execution1Output0 ;
    prov:startedAtTime "2018-09-18T10:52:00.802Z" ;
    prov:used repr:Cell1Execution1Source ;
    repr:executionTime "243ms" .

repr:Cell1Execution2 a repr:CellExecution ;
    p-plan:correspondsToStep repr:Cell1 ;
    prov:endedAtTime "2018-09-18T10:53:01.880Z" ;
    prov:generated repr:Cell1Execution2Output0 ;
    prov:startedAtTime "2018-09-18T10:53:01.703Z" ;
    prov:used repr:Cell1Execution2Source ;
    repr:executionTime "177ms" .

repr:Cell2Execution0 a repr:CellExecution ;
    p-plan:correspondsToStep repr:Cell2 ;
    prov:used repr:Cell2Execution0Source .
```

Figure 5.7: Provenance of Jupyter Notebook and its executions represented in RDF

we focus on helping the scientists to compare the results of different executions of a Jupyter Notebook.

We consider two use cases for the development of this module. They are:



```
In [15]: %matplotlib inline
         import matplotlib.pyplot as plt
         import numpy as np
```

```
In [16]: x1 = np.linspace(0.0, 4.0)
         y1 = np.cos(18 * np.pi * x1) * np.exp(-x1)
         plt.subplot(2, 1, 1)
         plt.plot(x1, y1, 'o-')
         plt.ylabel('Damped oscillation')
         plt.savefig("t1.png")
         plt.show()
```

Provenance Difference: Select the options to see the difference between two executions.

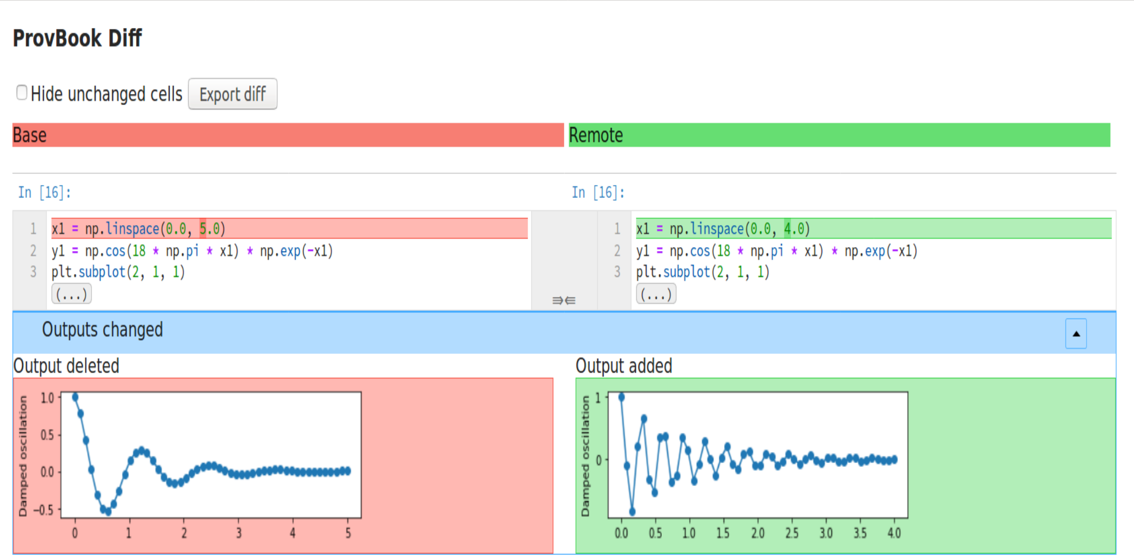
2018-07-11T11:40:59.199Z | 2018-07-11T11:40:59.199Z

```
In [17]: x2 = np.linspace(0.0, 5.0)
         y2 = np.cos(3 * np.pi * x2)
```

Figure 5.8: A Notebook code cell with the extension to compare its different executions.

- Repeatability

  Ana performs some data computational tasks using Jupyter notebook. She has a presentation and wants to show her results to her team members. Before her presentation, she wants to confirm her result by repeating her experiment.



ProvBook Diff

In [16]:

```
x1 = np.linspace(0.0, 5.0)
y1 = np.cos(18 * np.pi * x1) * np.exp(-x1)
plt.subplot(2, 1, 1)
(...)
```

In [16]:

```
x1 = np.linspace(0.0, 4.0)
y1 = np.cos(18 * np.pi * x1) * np.exp(-x1)
plt.subplot(2, 1, 1)
(...)
```

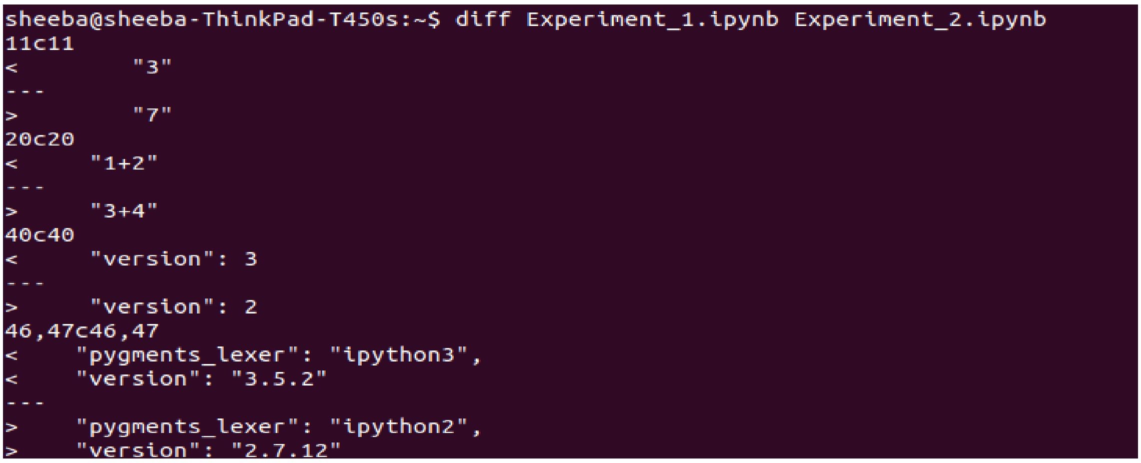
Figure 5.9: The difference between the input and output of two different execution of a code cell

She runs all the cells in the Jupyter notebook in the same laptop to confirm her results (see Repeatability Definition 4.1.5).

- Reproducibility
  Ana presents her result in her team meeting. She shares her Jupyter notebook to the team members. Alice, her team member, runs Ana's Jupyter Notebook in her own laptop which has a different operating system and memory specifications. She wants to compare the result she got from her execution of the notebook with Ana's result. She uses ProvBook to compare the two results from two different executions. Alice also makes some changes in the parameters and run the Jupyter Notebook in a non-linear order. She gets a different result and she wants to see what was the divergence that resulted in a different output (see Reproducibility Definition 4.1.4).

In the first use case, the computational experiment is repeated in the same environment by the same experimenter. While in the latter, the experiment is reproduced in a different environment by a different experimenter. In either case, the difference between the executions help the users either to (1) repeat and confirm/refute their results or (2) reproduce and confirm/refute others results.

ProvBook tries to address both the issues by providing a provenance difference module to compare the different executions of a notebook. Figure 5.8 shows a notebook code cell with the extension to compare its different executions. The start time of different executions collected in Section 5.2.1 is used to differentiate between two executions. The user is provided with a dropdown to select two executions based on the starting time of the executions. When the user selects the two executions, the difference in the input and the output of these executions are shown side by side.



```
sheeba@sheeba-ThinkPad-T450s:~$ diff Experiment_1.ipynb Experiment_2.ipynb
11c11
<         "3"
---
>         "7"
20c20
<      "1+2"
---
>      "3+4"
40c40
<      "version": 3
---
>      "version": 2
46,47c46,47
<    "pygments_lexer": "ipython3",
<    "version": "3.5.2"
---
>    "pygments_lexer": "ipython2",
>    "version": "2.7.12"
```

Figure 5.10: The diff of two notebooks using traditional diff tool

The users can select the original experimenter's execution with their own execution of the Jupyter Notebook as well.

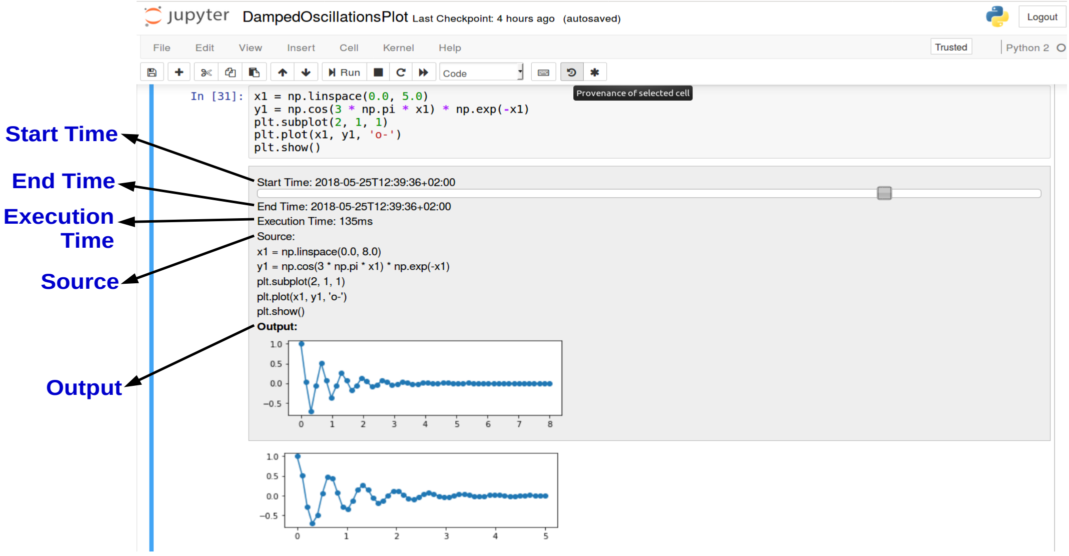
Figure 5.9 shows the differences between the input and output of two different execution of a code cell. If there are differences in the input or output, the difference is highlighted for the user to distinguish the change. As seen in Figure 5.9, there is a difference in the source in Line 1 in the two executions which is highlighted and has resulted in different outputs. The provenance difference module uses the nbdime[8] library from the Project Jupyter. The nbdime tools provide the ability to compare notebooks and also a three-way merge of notebooks with auto-conflict resolution. ProvBook extends the nbdime library and calls the API from the nbdime to see the difference between the provenance of two executions of a notebook code cell. Jupyter Notebooks are stored in a JSON file format which makes easy parsing because of its structure. Figure 5.10 shows the diff of the Jupyter Notebooks using traditional line-based tools. Since these tools do not handle the logical structure of the notebooks, nbdime is developed taking into account the structure of the notebook. It provides diffing of notebooks based on the content. It uses existing tools for the input and output and renders image-diffs properly. The current algorithm used by nbdime for the diffing is the Longest Common Subsequence [Hirschberg, 1977]. There is an ongoing work[9] to replace the brute force O(N^2) LCS algorithm with the Myers LCS based diff algorithm [Myers, 1986].

5.2.4 Provenance Visualization

Figure 5.11 shows a Jupyter Notebook code cell with the provenance data of its executions. The visualization of the provenance information is displayed below the input of every cell. A slider is provided in the provenance area where the user can drag to view the history of the executions of the cell. The user can track the history

[8] https://github.com/jupyter/nbdime

[9] https://readthedocs.org/projects/nbdime/downloads/pdf/latest/



```
In [31]: x1 = np.linspace(0.0, 5.0)
         y1 = np.cos(3 * np.pi * x1) * np.exp(-x1)
         plt.subplot(2, 1, 1)
         plt.plot(x1, y1, 'o-')
         plt.show()
```

Start Time: 2018-05-25T12:39:36+02:00
End Time: 2018-05-25T12:39:36+02:00
Execution Time: 135ms
Source:

```
x1 = np.linspace(0.0, 8.0)
y1 = np.cos(3 * np.pi * x1) * np.exp(-x1)
plt.subplot(2, 1, 1)
plt.plot(x1, y1, 'o-')
plt.show()
```

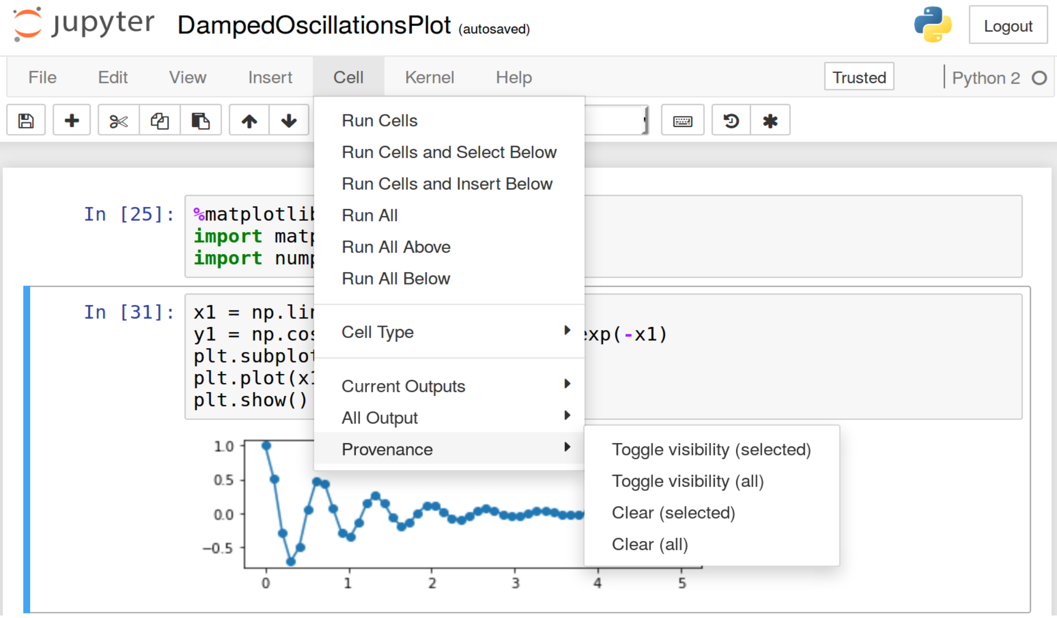
Output:

Figure 5.11: A Jupyter Notebook code cell with the provenance data of its executions

and compare the current results with several previous results and see the difference that occurred. The user can view the provenance information of a selected or all cells by clicking on the respective buttons in the toolbar. ProvBook also provides the user the options to clear the provenance information of a selected cell or all cells if needed. It tries to address the problem of having larger provenance information than the original notebook data. Storing the historical data in the notebook itself helps in easy portability. ProvBook adds a provenance menu in the Jupyter Notebook interface as shown in Figure 5.12. A user can toggle the provenance display for a selected cell from Cell → Provenance → Toggle visibility (selected). A user can clear the provenance data from the metadata of the notebook from Cell → Provenance → Clear (all).

5.3 Semantic Representation of Scripts

Scripts are widely used in computational experiments. They have become a vital part of the research lifecycle of experiments for scientists for automation, measurement, and analysis of data. The basic programming course provided by many universities helps the researchers to learn scripting languages. The ability to do complex tasks through minimal steps using scripts provide an added value in their research work. Scripts can be executed in several trials with different parameters for the analysis of data with less effort. However, the provenance information of the several executions of script are lost when they are re-executed. The importance for provenance management of Jupyter Notebooks presented in Section 5.2.1 applies for the execution of scripts as well. To represent complete path of a scientific experiment, it is important that we link the retrospective provenance with the prospective



```
In [25]: %matplotlil
         import matp
         import nump

In [31]: x1 = np.lir
         y1 = np.cos                        xp(-x1)
         plt.subplot
         plt.plot(x1
         plt.show()
```

Figure 5.12: The provenance menu added in the Jupyter Notebook interface

provenance. In this section, we present our work on semantic representation of script provenance. One of our aims to do so is to link the non-computational data with the computational results to get a complete path of a scientific experiment. Reproducibility of these scripts is thus an important step towards reproducibility of the experiments as a whole.

To semantically represent the script provenance, we first need to understand the important components required for computational reproducibility of scripts. Here, we do not focus on one particular scripting language instead focus on the general structure of scripts. We present the components that we consider are important to know while reproducing others' script. They are as follows:

SC1 Script

This is the program or code that is used in a scientific experiment. It denotes the prospective provenance specifying the functions that need to be executed.

SC2 Function

It is a programming language code snippet in a script which describes a procedure or a routine. It takes zero or more input and returns zero or more output to perform some tasks.

SC3 Module

It is a part of a computer program or software which provides declarations and functions. Programs can contain one or more independently developed modules. Most of the scripts start with importing modules. *ModuleNotFound* is a very common error during the script execution.

SC4 Module Version
The version of modules is a very important part of script execution. The version can affect the intermediate and final results of a script.

SC5 Argument
This is the parameter taken as an input, or declared/used in a script. This is an important provenance information since the output value can depend on it.

SC6 Input
It is the variable used as an input to a script or a function.

SC7 Output
It is the variable generated as an output of a script or a function.

SC8 Programming Language
It is the programming language in which a script is written.

SC9 Programming Language Version
It is the version of the programming language in which a script is written.

SC10 Operating System
It is the operating system where the script is run.

SC11 Operating System Version
It is the version of the operating system where the script is run.

SC12 Author
It denotes the person who is the author of the script.

SC13 Function Activation
It denotes when a function is activated or run.

SC14 Trial
It denotes a run or execution of a script.

SC15 Start Time
It denotes the time when the script is started to execute.

SC16 Finish Time
It denotes the time when the script finishes its execution.

SC17 Experimenter
It denotes the person who is executing the script.

SC18 Location
It denotes the location where the script is executed.

SC19 Accessed File
It denotes the files that are accessed during the script execution.

SC20 Order of execution
It denotes how the functions are executed inside a script.

SC21 Experiment
It denotes the scientific experiment in which the script was used to perform data computation to produce result.

We have presented the components of a script that are required for computational reproducibility. We use each of them to semantically describe the provenance of the complete execution of a script in a structured form using linked data without worrying about any underlying technologies or programming languages. We use the REPRODUCE-ME ontology extended from PROV-O and P-Plan to to describe the steps and sequence of steps in the execution of a script. Before extending the ontology to describe script provenance, we define the competency questions as follows:

CQ20 What is the sequence of steps in the execution of a script with input parameters and intermediate results in each step required to generate the final output?

CQ21 Which are the steps that invoke a particular module?

CQ22 Which are the environmental attributes in the execution of a script?

CQ23 List the user, the operating system, programming language version, the working directory associated with the execution of a script.

CQ24 What is the complete derivation of a script output?

To answer these questions, it is required to know which input data was responsible for the output, the steps involved in generating them, the functions, the input parameters involved, the dependencies to other modules, time taken for the execution of each function, the side effects, etc. Table 5.1 describe how each component (SC1-SC21) is modelled in the REPRODUCE-ME ontology. Each ontology term is categorized into prospective and retrospective provenance (see Section 1.2). Prospective provenance of a script denotes the specification and the steps required to follow to generate the results. While, retrospective provenance of a script denotes what actually happend when the script was executed. The prospective provenance in the context of script execution includes script, function, module, programming language in which the script is written, author, and experiment. The version of the module, programming language, and operating system belongs to retrospective provenance since they can change in each execution of the script. This is the same

Component	**Ontology term**	**Provenance**	**Remarks**
SC1	*repr:Script*	Prospective	Subclass of *p-plan:Plan*
SC2	*repr:Function*	Prospective	Subclass of *p-plan:Plan*
SC3	*repr:Module*	Prospective	Subclass of *p-plan:Plan*
SC4	*repr:Version*	Retrospective	Subclass of *repr:Setting*
SC5	*repr:Argument*	Retrospective	Subclass of *p-plan:Variable*
SC6	*repr:Input*	Retrospective	Subclass of *p-plan:Variable*
SC7	*repr:Output*	Retrospective	Subclass of *p-plan:Variable*
SC8	*repr:ProgrammingLanguage*	Prospective	Subclass of *repr:Setting*
SC9	*repr:Version*	Retrospective	Subclass of *repr:Setting*
SC10	*repr:OperatingSystem*	Retrospective	Subclass of *repr:Setting*
SC11	*repr:Version*	Retrospective	Subclass of *repr:Setting*
SC12	*repr:Author*	Prospective	Subclass of *prov:Person*
SC13	*repr:FunctionActivation*	Retrospective	Subclass of *p-plan:Step*
SC14	*repr:Trial*	Retrospective	Subclass of *prov:Activity*
SC15	*prov:startedAtTime*	Retrospective	Data property
SC16	*prov:endedAtTime*	Retrospective	Data property
SC17	*repr:Experimenter*	Retrospective	Subclass of *prov:Person*
SC18	*prov:Location*	Retrospective	Using *prov:atLocation*
SC19	*repr:File*	Retrospective	Subclass of *p-plan:Variable*
SC20	*p-plan:isPrecededBy*	Retrospective	Object property
SC21	*repr:Experiment*	Prospective	Subclass of *p-plan:Plan*

Table 5.1: Overview of the ontology terms to model script provenance

case with the input, output, argument, function activation, start and end time, file, and order of execution. Figure 5.13 shows the modelling of script provenance using REPRODUCE-ME Ontology.

We describe how adding these important concepts in REPRODUCE-ME ontology help in addressing each competency questions.

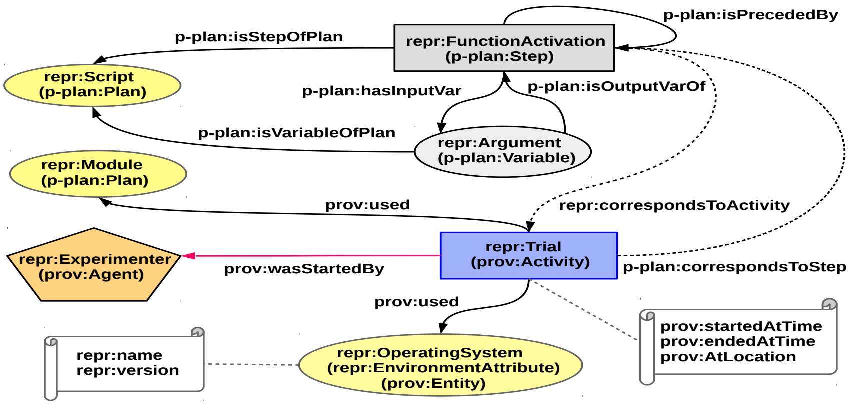


Figure 5.13: The REPRODUCE-ME Ontology depicting the script execution

CQ20 What is the sequence of steps in the execution of a script?
The concept *Trial* is added to model the execution of a script. Each *Trial* of a script results in several activation of functions. So the concept *FunctionActivation* is modelled as a *p-plan:Step*. Each *FunctionActivation* is linked to the *Trial* with the object property *p-plan:correspondsToStep*. The order of the script execution is modeled using the object property *p-plan:isPrecededBy* where every *FunctionActivation* is preceded by another *FunctionActivation*.

CQ21 Which are the steps that invoke a particular module?
Each *FunctionActivation* corresponds to a *Trial*. The concept *Module* is added to model the module. Each *Trial* is linked to *Module* using the object property *prov:used*.

CQ22 Which are the environmental attributes in the execution of a script?
Several environmental attributes affect the execution of a script. The concept *EnvironmentAttribute* represents the class of the environmental attributes of a scientific experiment. We have added the specific concepts like *ProgrammingLanguage*, *OperatingSystem*, *Version* to capture execution environment provenance of scripts.

CQ23 List the user, the operating system, programming language version, the working directory associated with the execution of a script.
The concepts *Author* and *Experimenter* are added as subclass of *prov:Person* to represent the author and experimenter who executes the script respectively.

The concepts *ProgrammingLanguage*, *OperatingSystem*, *Version* and *Location* are used to model the programming language, operating system, the version of programming language and operating system, and the location where the script is executed respectively.

CQ24 What is the complete derivation of a script output?
The complete derivation of a script output consists of a path from the input to the output. This path provides information about the script used, the functions defined, the functions which were activated, the trial of the script, the execution time of the trial (start and end time), the modules which were used and their version, the programming language of the script and its version, the operating system where the script is executed and its version, the files that were accessed during the execution, the input argument and return value of each function activation and the final result. Table 5.1 provides the ontology terms that are added to represent this complete path. Figure 5.13 shows how these terms are used to model the script execution.

5.4 Summary

This chapter presented approaches to support computational reproducibility. We saw that the computational notebooks and scripts are widely used in computational environments. The reasons for their wide adoption include ability to perform complex tasks with minimum effort, easy to learn, use, deploy and share. We first focused on the provenance management of computational notebooks and later on the scripts. We presented the end-to-end provenance management for computational reproducibility. Provenance capture, representation, storage, query, difference, and visualization are the important modules for the end-to-end provenance management for computational reproducibility. We showed how we developed each module for provenance tracking for computational notebooks. We introduced ProvBook, which is an extension of Jupyter Notebooks which captures the provenance of their executions. The three important modules in ProvBook help to capture, represent and compare the provenance of their executions. We also showed how we semantically describe the computational notebooks and scripts along with their executions. We evaluate ProvBook with respect to different scenarios and answer the competency questions related to computational experiments in Chapter 7. The results of the evaluation are provided in Section 7.4.

Chapter 6

CAESAR-A Collaborative Environment for Scientific Analysis with Reproducibility

As stated in Chapter 2, one of our goal is to design and create a provenance-based semantic framework to collect information about the experimental data and results along with the settings and execution environment and visualize the complete path (Goal3). We have developed a conceptual model using semantic web technologies to describe a complete path of a scientific experiment in Chapter 4. We used this model to semantically represent the computational notebooks and scripts and their execution in Chapter 5. We also showed how ProvBook captures the provenance of computational experiments using Jupyter notebooks. In this chapter, we aim to integrate the contributions from Chapter 4 and 5 to provide a framework for the scientists to capture, represent, store, query, compare and visualize the complete path of a scientific experiment consisting of both computational and non-computational steps.

To describe the complete path of a scientific experiment, we need end-to-end provenance management support in scientific data management platforms. Provenance capture, representation, storage, difference, and visualization are the core units of end-to-end provenance management systems (see Figure 5.1). Each unit plays a major role here thus supporting understandability, reproducibility, and reuse. However, the lack of end-to-end provenance management support in such platforms is currently challenging (Chapter 3).

The rapid increase in the volume, variety, and complexity of research data in recent years brings several challenges in their management. The decision on which data to keep and at what granularity are some of them. It is also often difficult to follow a relation between the results in a publication and the steps that generated them. Not only the results and the steps but also all the data elements that are essential for reproducing results (see Section 4.1) are required to follow this link. It helps in building up the trust and confidence in results. It is also important that the datasets along with the metadata are collected and organized in a structured way from the beginning of the experiments. Therefore, we need to start addressing this issue at the stage when the data is created (see Figure 1.1). Thus, scientific research data management needs to start at the bottom level of the research lifecycle to play a key role in this context.

To support our main hypothesis that it is possible to capture, represent, manage and visualize a complete path taken by a scientist in an experiment including the computational and non-computational steps to derive a path towards experimental results (Section 2.3), we aim to help the scientists at the grass root level not only to describe (see Chapter 4) but also manage their experimental data in a reusable and interoperable way. In addition to that, we intend to provide a collaborative environment where scientists can view, share and reuse each others' experimental data. Visualization of big data provenance for understanding the derivation path of the results is another requirement in this context. It helps the newcomers in a research team to understand and visualize the experiments conducted by their team members with minimal effort. Therefore, we aim to provide a platform which captures and manages the scientific experiments along with the input, output and execution environment and link with other datasets on the web using the Semantic Web technologies. This platform provides a better understanding of these experiments with the help of visualization techniques and ontologies.

With the requirements defined in Section 2.5, we present CAESAR (**C**oll**A**borative **E**nvironment for **S**cientific **A**nalysis with **R**eproducibility). It provides a collaborative environment for authoring of scientific experimental metadata. It provides all the core units required for the end-to-end provenance management. The complete path of an experiment including its computational and non-computational parts is provided with the provenance representation and visualization. We also explain how we address each scenario faced by Ana that we presented in our use case in Section 2.1 using CAESAR. The code of CAESAR is available online[1].

In this chapter, we introduce CAESAR in Section 6.1 and its underlying architecture in Section 6.1.2. The following sections discuss in detail each module of CAESAR for the end-to-end provenance management of scientific experiment as shown in the Figure 5.1. Section 6.2 presents the features of the provenance capture module. Section 6.3 presents the provenance data management module. How the provenance is represented in CAESAR is described in Section 6.4. The visualization modules of CAESAR are discussed in Section 6.7. The implementation details are explained in Section 6.8. This chapter concludes with the summary (Section 6.9). Parts of the results of this chapter have been published in [Samuel et al., 2017,Samuel et al., 2018].

[1] `https://github.com/CaesarReceptorLight`

6.1 CAESAR: An Introduction

To support reproducibility and reuse of scientific experiments, we narrowed our scope of research data management to life sciences. While the research data comes from multiple contexts in life sciences, we focus on the field of light microscopy imaging. This is due to the fact that research in life sciences is mostly based on imaging datasets. These imaging datasets capture the spectral characteristics of the signals generated from the samples to measure and understand the functions of the cells and tissues of organisms and plants. These datasets generated from different imaging methods often have proprietary file formats and can be viewed with only particular hardware and software. The lack of a standardized file format for the imaging datasets is a fundamental problem in this area [Allan et al., 2012]. Explosive growth in the number of biological images and their sheer size also make their management and querying challenging.

To overcome these challenges, the Open Microscopy Environment was started as an international effort of universities and industries to build open source tools and standards for microscopy imaging data. We conducted a literature survey to find a suitable open source imaging data management. The survey focused on two platforms: BisQue [Kvilekval et al., 2010] and OMERO [Allan et al., 2012].

The Bio-Image Semantic Query User Environment (BisQue) is an open-source server-based software system that can store, display and analyze images. The stored images can be accessed through a web interface or by using an API. It is being developed and maintained by a small team at UCSB. They have two releases per year scheduled. The platform uses the Bio-formats[2], OpenSlide[3], and ImarisConvert[4] to support over 240 file formats.

OMERO [Allan et al., 2012] is another open source data management platform for imaging metadata primarily for experimental biology. The OMERO software platform is developed by the Open Microscopy Environment (OME) which is a collaborative consortium responsible for producing open specifications and tools to enable open-access of image data. Its plugin architecture provides a rich set of features including analyzing and modifying images. It supports over 140 image file formats using BIO-Formats [Linkert et al., 2010]. OMERO has a very active development community ensuring a continued effort to improve the system, with everybody being able to contribute. It has also a well-documented API to write own tools and the ability to extend the web interface with plugins. It also profits from a faster release cycle.

To support reusability of software, we decided to select a suitable data management

[2] https://www.openmicroscopy.org/bio-formats/

[3] https://openslide.org/

[4] http://www.bitplane.com/

from these two tools. Both the software provide more or less the same features for the image data management. But neither of these tools provided the management of the provenance of experimental data including the computational processes. We selected OMERO based on the possibility of its rich and extensible features. However, the lack of semantic representation of experiments with the integration of data and results from different steps and sources led to the development of CAESAR.

CAESAR is developed for the data management of experimental datasets and its provenance [Samuel et al., 2017] as part of the CRC ReceptorLight[5]. It is a software platform which is extended from OMERO. Together with the rich features provided by OMERO and our extensions[6], the CAESAR provides a platform to support understandability and reproducibility of experiments. It provides an added value with the semantic integration by providing the linking of the datasets with the experiments along with the execution environment. It gives the scientists the features to describe, preserve and visualize their experimental data along with the images [Samuel et al., 2018].

6.1.1 OMERO: Architecture

To understand CAESAR, it is important to understand the architecture and the features provided by the underlying software, OMERO. Figure 6.1 shows the architecture of OMERO. It is composed of a server and clients which are written in Java, Python, and C++ [Allan et al., 2012]. It is a collection of databases, middleware, and clients for the management and processing of images. The main component is the OMERO.server which is responsible for connecting the databases that store different data types. It provides access to the storage data to the client application using a single API. It consists of several databases to store heterogeneous data types. The data including thumbnails, images, binary data and the data used for annotation are stored in a flat file store provided by the "Binary Repository". It also stores the scripts and other files attached along with the images. Text indexing in OMERO is provided by Lucene[7] and the indices are stored in "Search Index". The relational database is provided by PostgreSQL. The metadata associated with the images and the annotations are stored in the relational database. The HDF5-based tabular data provided by OMERO.tables store all the table-based data. The images are stored as binary pixel file in its file repository. The proprietary file format of the

[5] `http://www.receptorlight.uni-jena.de/`

[6] Daniel Walther, Frank Taubert and Sheeba Samuel contributed to the implementation. Role of Daniel Walther is in the extension of OMERO.server to include new services, Frank Taubert in the development of the desktop client, and Sheeba Samuel in the development of the webclient plugins and the semantic integration.

[7] `http://lucene.apache.org/`

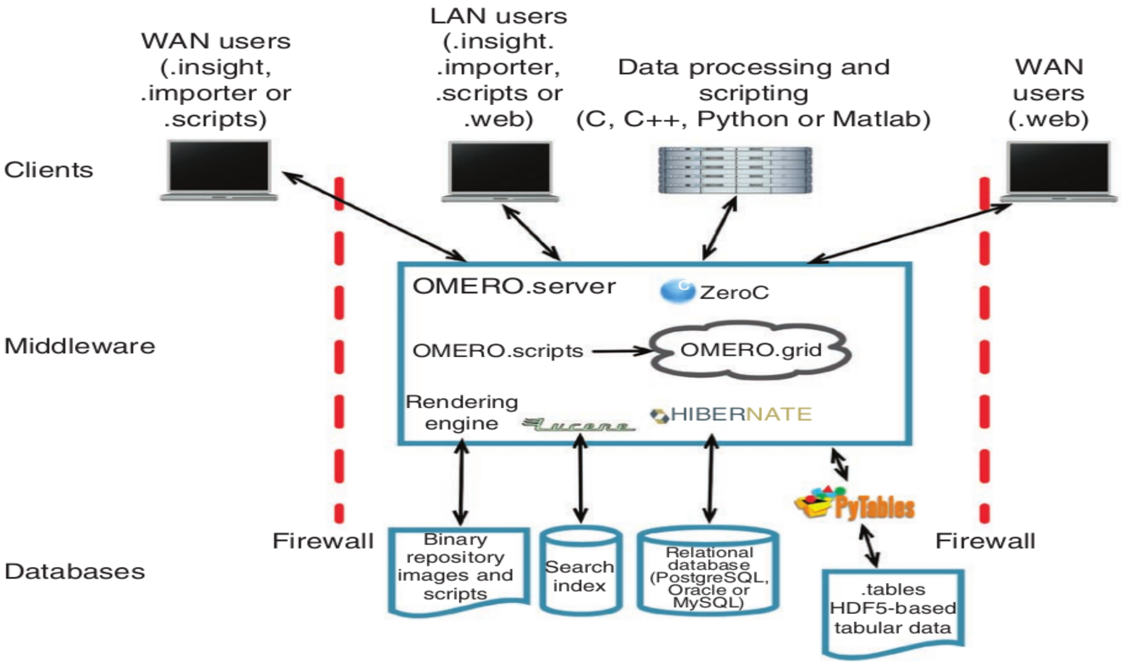

WAN users (.insight, .importer or .scripts)
LAN users (.insight. .importer, .scripts or .web)
Data processing and scripting (C, C++, Python or Matlab)
WAN users (.web)
Clients
Middleware
Databases
OMERO.server
ZeroC
OMERO.scripts
OMERO.grid
Rendering engine
Lucene
HIBERNATE
PyTables
Firewall
Firewall
Binary repository images and scripts
Search index
Relational database (PostgreSQL, Oracle or MySQL)
.tables HDF5-based tabular data

Figure 6.1: The architecture of OMERO [Allan et al., 2012]

images is converted to a key-value pair in a table. These are added as an annotation to the image in OMERO.

The "Rendering Engine" component in the middleware is responsible for reading the image data and renders it based on the particular client application. Search Queries are performed by Lucene based on the search indices stored in the "Search Index". Scripting service provided by OMERO.scripts is used for processing images using scripts. Hibernate provides the relational mapping between the relational database and the OMERO.server. The Zero's Internet Communications Engine (ICE) provides the communication to different clients using the single OMERO API. This API provides access to the data from the Binary Repository, relational databases, and HDF5 files.

OMERO provides a Python-based web client which is one of the important user interfaces. Using the webclient, users can visualize, analyze, annotate and share the images. But the webclient does not provide the feature to upload images which is considered as a major drawback. The uploading of images is possible only through the OMERO.insight which is a Java-based client. OMERO.importer which is written in Java reads and extracts the image acquisition data with the help of BIO-Formats. BIO-Formats [Linkert et al., 2010] is an image translation library which reads and converts the proprietary microscopy data to an open standard model so that they can be used by other tools.

The OME provides OME data model which describes the elements responsible for the image acquisition process in a microscope. The OMERO is based on the OME data model and BIO-Formats which helps to manage the heterogeneous image data. The desktop client and a webclient help the users to manage their data. Image

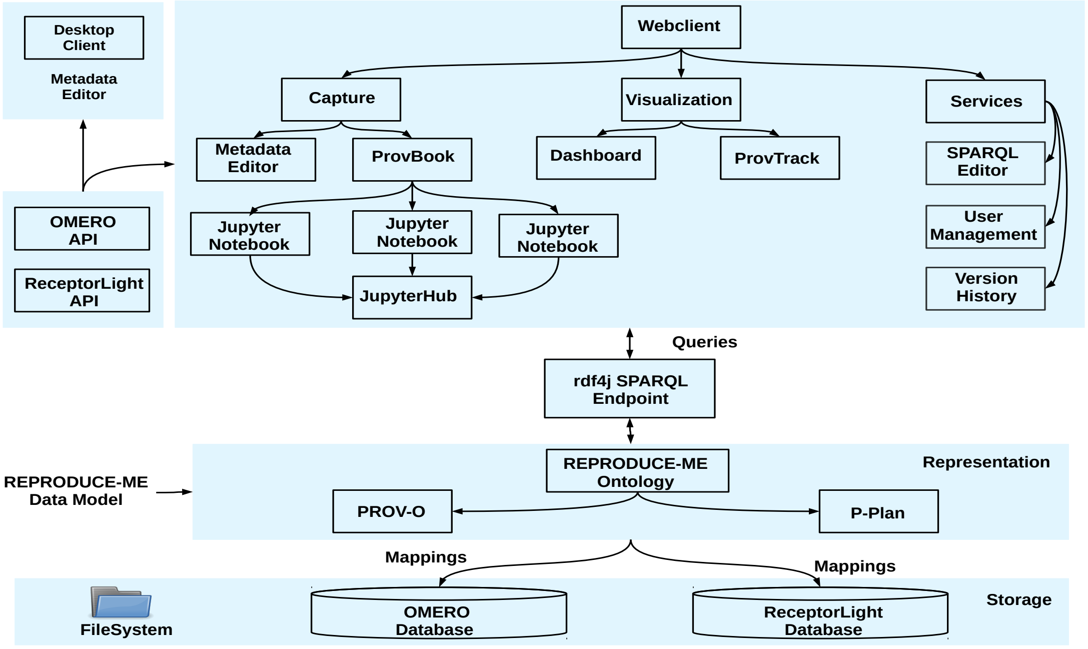


Figure 6.2: The architecture of CAESAR

data are uploaded to OMERO using the desktop client and mapped to the OME data model. OMERO provides the visualization of images. The imported images can be further analyzed to derive a new set of images. These images can be stored along with the original image. It provides a feature called "StructuredAnnotations" where files like measurements can be added and stored in the file repository. The annotations can be linked with the ontologies. The original and the derived datasets can be accessed using the OMERO API.

6.1.2 CAESAR: Architecture

The architecture of CAESAR is built on top of OMERO and BIO-Formats. The plugin architecture provided by OMERO allows a modular design to add new applications. The aim of CAESAR is to provide the data management of scientific experimental data and its provenance to support reproducibility and reuse. To provide the complete path of a scientific experiment, it is necessary that CAESAR provides the facility to capture, represent, store, query, compare and visualize the provenance information of experiments. Figure 6.2 shows the architecture of CAESAR. Hence, the architecture of CAESAR focuses on the following modules:

M1 Provenance Capture: This module is the primary component for capturing provenance of experiments. This module is implemented in the webclient as a separate plugin. The experimental metadata is captured using the *Metadata Editor* while the provenance of computational experiments is captured using ProvBook in a multi-user environment provided by JupyterHub connected to CAESAR (Section 6.2).

M2 Provenance Storage: This module is responsible for the storage of the provenance of experiments. In addition to the storage provided by OMERO, the RDF data provided by the mapping of the REPRODUCE-ME ontology and the relational databases are queried using the rdf4j SPARQL Endpoint[8] (Section 6.3).

M3 Provenance Representation: This module provides the semantic description of experiments. The link between the experimental metadata, data, steps, settings and the results is semantically described using the REPRODUCE-ME ontology. The ontology-based data access is used to represent the mapping between the underlying OMERO and ReceptorLight database (Section 6.4).

M4 Provenance Visualization: This module provides the visualization of the complete path of scientific experiments. The Project Dashboard provides a complete overview of the experiments performed in a research project. Whereas the ProvTrack provides an interactive provenance graph to track the provenance of each scientific experiment (Section 6.7).

Each module is discussed in detail in the following Sections 6.2-6.7. The chapter is concluded with a summary (Section 6.9).

6.2 CAESAR: Provenance Capture

The first and the foremost important step towards reproducibility is the capturing of provenance. Lack of documentation and digitalization of experimental data, lack of data integration from different devices, and publicly available data hinder research reproducibility. In order to avoid the above-mentioned problems, we provide a way for scientists to capture provenance of experimental data. Often, it is difficult for scientists to learn an entirely new system. The Provenance Capture module provides a metadata editor with a very rich set of features to describe the experimental metadata with the ease of writing in their lab notebook.
The Provenance Capture Module is the primary module of CAESAR to capture the experimental metadata. The main component of this module is a metadata editor which is a form-based provenance capture system. The editor helps the scientist to document their experimental metadata and interlink with other experiment databases. The requirements to design the provenance capture module were collected from the scientists from the CRC ReceptorLight project. The administrators of each research project are responsible for creating the template of the form. This is to ensure that the scientists in a research project capture the metadata of experiments in a uniform manner. The general template consists of an "Experiment"

[8] `http://rdf4j.org/`

form which documents the information about an experiment. It includes the temporal and spatial properties as well as the research context of the experiment. The materials and other resources used in an experiment are added as new templates to be included in CAESAR. The template is then added as a service and a database table in CAESAR.

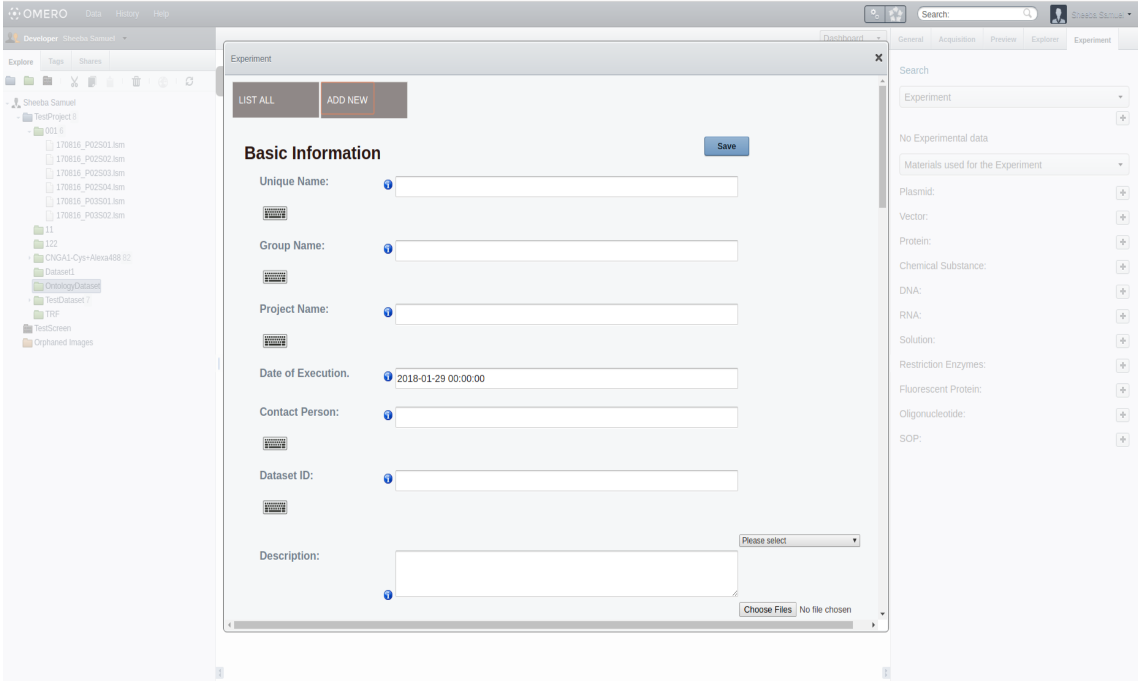
To add new metadata support, OMERO.server is extended to include new services. The REPRODUCE-ME data model is used to describe scientific experiments and their provenance. Every time new data types are added in CAESAR, it is difficult to update the whole application by hand. To ease this, new data types are added to the OME Acquisition XML file. The code generators are constructed which reads the OME-XML file. These are converted into scripts which creates the ReceptorLight database and links to the OMERO relational database. The code that is generated through this process is compiled which is later used by the webclient to capture experimental metadata. These new data types are also added to ICE (see Section 6.1.1) so that the new data types are available as API to be used by remote clients. This approach helps to update and maintain new data types which is commonly agreed by members in a research project. Also, it helps the development process to avoid being error-prone.

The webclient and desktop provided in CAESAR are developed to include the new data types. The desktop client was developed to deploy the new system on workstations where an Internet connection is not available. This is required due to the security reasons in some of the microscopes in the laboratory. So, when a researcher is conducting an experiment, he or she can input all the data in the desktop client [Samuel et al., 2017], thus minimizing the loss of data naturally occurring when recording these things from memory. The user can then upload the images, files, and measurements obtained from the devices during the experiment which is later uploaded to the server when an Internet connection is available. The provenance capture module in the desktop and webclient is similar as well as their user interfaces to maintain consistency across clients.

Using the metadata editor, the scientists can easily record all the data of the non-computational steps performed in their experiments. In addition to the experiment, the plugin allows documenting the protocols, the materials and the steps that were used. Figure 6.3 shows the Experiment Metadata Editor. We discuss in detail the additional features provided by the Experiment plugin in CAESAR.

1 ***User and group management for the experimental data***

The user and group management is important if a data management system has to be used by many people and communities. OMERO offers a detailed way to manage users in groups and provides roles for these users. A group is a collection of users which enables sharing of data between them. The roles and permissions are assigned to the users belonging to a group to restrict the



Basic Information

Unique Name:

Group Name:

Project Name:

Date of Execution. 2018-01-29 00:00:00

Contact Person:

Dataset ID:

Description:

Figure 6.3: The Metadata Editor in CAESAR

modification of data.

The user and group management provided by OMERO is also adopted in CAESAR. A user may belong to one or many groups. The data is shared between the users in the same group in the same CAESAR server. The data can be made available to members of other groups based on the permission level of the group. The data imported by the user belongs to the group where the user has documented the data. The data can also be moved to other groups. There are three roles a user can be assigned:

- *Administrator*
  A server can have multiple administrators. The administrators control all the settings of the groups.
- *Group Owner*
  Each group can have multiple owners. The group owner has more rights than a group member within a group. The group owner has also the right to add other members to the group.
- *Group Member*
  A group member is a standard user in the group.

In addition to the roles, there are also various permission levels in the system.

- *Private*
  This is the most restrictive permission level. A private group owner can view the members of the group. The owner can also view and control the data of the members within a group. Whereas a private group member

can view and control only his/her own data. This permission level has the least collaboration level with other groups in the system.

- *Read-only*
  This is an intermediate permission level which allows viewing the members and read their data. The group owners can read and perform some annotations on the data of members from other groups in addition to their own group. While the group members can view the members from other groups and read their data, they don't have permission to annotate the datasets.
- *Read-annotate*
  This permission level provides a more collaborative option. The group owners and group members can view the members of the other groups as well as read and annotate their data.
- *Read-write*
  This permission level allows all group members to read and write data just like their own group.

CAESAR also uses the same role and permission levels to control the access and modification of experimental data. In a *private* group, a Principal Investigator (PI) can act as a group owner and students as group members. The students can store their experimental data and PIs can access students' data and decide which data can be used to share with other collaborative groups.
In a *Read-only* group, a scientist can move data to a read-only group so that the data can be viewed by other group members. The group owner can then annotate their data or add Regions of Interests to their images. This group can serve as a public repository where the original data for the publications are stored.
A *Read-annotate* group can serve as a collaborative team of groups who work together with the data for a publication or research. A *Read-write* group works in a very collaborative way where every group members are trusted and given equal rights to view and access the data.
It is possible for the group owners and administrators of the server to change the permission level of a group. But only the administrator can promote a group to Read-Write group. Also, it is not possible to degrade a groups' level to private if the data contain annotations made by other users. Users can perform the following actions based on the permission level in their group:

- Create Project and Datasets
- Upload Images and data
- Delete Data

- Move data between groups, projects, and datasets
- Run Scripts
- Use Regions of Interest (ROIs) (add, import, edit, delete, save and analyze them)
- Annotate, rate and tag images, add attachments and comments
- View and edit experimental data
- Reference experimental materials in other experiments

This user and group management opens the doors for collaboration among teams in research groups and institutes before the publication of the data online. This addresses one of the issue faced by Ana in the use case in Section 2.1. In this way, Ana could share her experimental data along with her team members and her supervisor. To share data with her collaborating team in another location, she could give the other team permission to view and use the data.

2 ***Link experiments with materials and resources***

A scientific experiment consists of computational and non-computational steps and processes. The non-computational processes in life sciences use several experiment materials and samples. In addition to that, it consists of input files, measurement files, and images. Each step of an experiment uses different materials and standard operating procedures. It is essential to interlink these dependencies of these materials with an experiment. To do so, the plugin provides the user with a facility to link the materials to the steps of an experiment. The input field in these forms is provided with an additional field so that the user can choose the resources from other tables in the database. Also, the user can attach files, scripts or other resources to any steps of an experiment form. These resources can either be an input to a step or intermediate result of a step. The users can also add the publications that were used as a reference for the experiment. The files that are attached to these forms are stored in the Managed Repository of that particular user.

3 ***Reuse of experiment materials and Standard Operating Procedures***

Reuse is an important factor when working in a collaborative research team. In a group, it is important to reuse things than doing it from scratch. Reuse in CAESAR is achieved by sharing the descriptions of the experiments, standard operating procedures, and materials with the team members within the research group. This avoids the need for documenting it multiple times. It is possible by referencing these descriptions in their own experiment. The plugin provides a database of experiment materials like Plasmid, Protein, Vector, etc.

The users in a research team can view the list of all the databases used by the other team members in their research group. The materials used in the group are visible to all the members of the group. Therefore, the scientist can use the description of materials and standard operating procedures used in his/her experiment.

4 ***Version history of experimental data***

Version management plays an important role in data provenance. Keeping the version history of the changes in the description of experiments helps in data provenance. In a collaborative environment, it is necessary to know the modifications made by the members of the system. Also, it is important to track the history of the outcome of an experiment. CAESAR provides version management of the experimental metadata. It stores any changes made in the documentation of an experiment. It also provides a facility for the user to view the version history of an experiment and compare two different versions of an experiment description.

5 ***File management***

The input data, measurement data or other resources which are attached as files to the experiment are stored in the Managed Repository of the server. The file management system in CAESAR store these files and index them to the experiment. The user can also organize the files in a hierarchical structure based on their experiments and measurements.

6 ***Standard Operating Procedures***

Each experiment has multiple non-computational and computational steps. A Standard Operating Procedure (SOP) in life-sciences provides a set of step-by-step instructions to carry out a complex routine. CAESAR provides a database of Standard Operating Procedures. The users can store in this database the protocols, procedures, scripts or Jupyter Notebooks based on their experiments. Later, these procedures can be linked to the step in an experiment where they were used. Users can reuse the SOPs created by other members as well. This database of SOPs can also contain scripts that were used to analyze data or images. Jupyter Notebooks can also be added which contains either the documentation of each step of the procedure or code for analysis. Some cells in the Jupyter Notebook just document the process but others may contain executable code which can be either code in R, Matlab or Python or Unix shell commands. Jupyter Notebooks helps to address various users irrespective of the programming languages they use. As the data and images are contained in the system itself, it is easy to include the scripts that analyze the data stored in the platform. The JupyterHub is installed along with ProvBook in CAESAR. This help users to execute the steps again to check whether the output is similar to the output from the previous users.

7 ***Provenance collection of executable steps***
The granularity of provenance collection is important to scientists. Sometimes, it is essential to know the fine details of the experiment workflow. Scripts are also part of some experiments. We collect the provenance information of the output generated by the Python scripts using noWorkflow tool. There are some sample scripts provided by OMERO to perform some operations in the images. The provenance of the execution of these scripts is captured and stored.

8 ***Annotate experiment with ontologies***
In addition to REPRODUCE-ME Ontology, the user can also annotate the experimental data with terms from other ontologies like GO [Ashburner et al., 2000], CMPO [Jupp et al., 2016], etc.

9 ***User proposals on experiment descriptions***
Based on the permission level of experimental data, if a user does not have the right to modify other member's data, then that user can propose changes to the experiment. This is done using the Proposal Feature provided by the prototype. The PIs from the current group or other groups can provide suggestions and modify the experimental data. The owner of the experiment receive those suggestions as proposals. The user has two options: First, to accept the proposal and add it to the current experimental data. Second, to reject the proposal and delete the proposal.

10 ***Metadata editor additional features***
To fasten the process of documentation of experimental data, we also provide autocompletion of data. In chemical databases, if the user provides the CAS number of the chemical, then the molecular weight, mass, structural formulas are fetched from the CAS registry and populated in the Chemical database. Similarly, for other materials like Protein, Plasmid, and Vector, the prototype provides additional data from the external servers. The DOI/PubMedId of the publications provided by the user helps to autofill the data about the authors and other publication details. In order to document descriptions with special characters, we provide a virtual keyboard for every input field. This helps the user to enter chemical formulas and symbols.

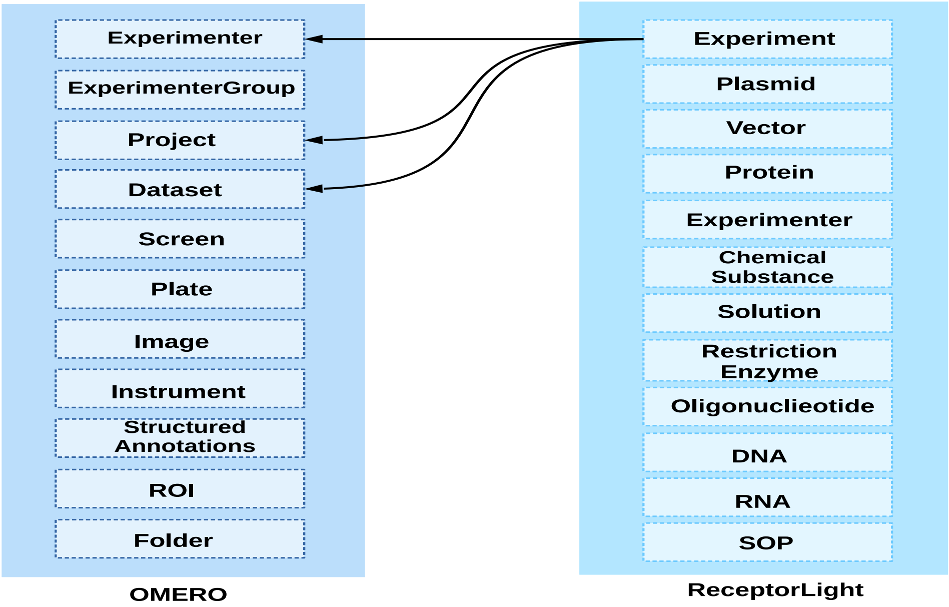


Figure 6.4: CAESAR Schema

6.3 CAESAR: Provenance Data Management

In this section, we discuss how the provenance information is stored in order to efficiently query this data from different sources. To understand the data management of provenance information, we discuss the database schema of CAESAR in general. The database used by CAESAR is PostgreSQL because of its underlying OMERO database. Figure 6.4 shows some aspects of the OMERO and ReceptorLight Schema which constitutes CAESAR. The important classes for the data management of images include Project, Dataset, Folder, Plate, Screen, Experiment, Experimenter, ExperimenterGroup, Instrument, Image, StructuredAnnotations, and ROI. A *Project* is a group of *Dataset*s. A *Dataset* is a collection of images which are generated for an experiment. A *Dataset* can belong to one or more *Project*s and a *Project* may contain one or more datasets. An *Image* is the actual image with its metadata. A *Dataset* can have more than one *Image*s and an *Image* can belong to one or more *Dataset*s.

An *Instrument* describes the device which is used to capture the *Image*. The *Instrument* model consists of *Microscope*, *LightSource*, *Detector*, *Objective* and *Filters* components. Each component of *Instrument* consists of elements which describes its *ManufacturerSpec* and *Settings*. The *Experimenter* describes the person who is performing the imaging experiment. The *StructuredAnnotations* consists of unordered collection of annotations that are attached to the objects like *Project*, *Dataset* or *Image*. The different types of *StructuredAnnotations* include *XMLAnnotation*, *FileAnnotation*, *ListAnnotation*, *LongAnnotation*, *DoubleAnnotation*, *CommentAnnotation*, *BooleanAnnotation*, *TimestampAnnotation*, *TagAnnotation*, *Ter-*

mAnnotation and *MapAnnotation*.

The *Experiment* class provided by OMERO database describes the type of experiment and the optional description field contain free text to further provide information about the experiment. But this will not suffice our requirements for capturing provenance information. Because of this limitation, we provide a separate data model to capture provenance information of the experiments.

The ReceptorLight database model consists of several important classes. Based on the data model discussed in Chapter 4, the schema is designed. The *Experiment* links all the provenance information together. The *Experiment* consists of temporal and spatial information. It describes the research group and the project. It connects all the images together which were captured during an experiment. An *Experiment* belongs to only one *Dataset* and has a one-to-one relationship. It links the several steps that were conducted in the experiment with its description. The data model also consists of several classes which are the materials used in the experiment. The model consists of *Plasmid*, *Protein*, *Vector*, *ChemicalSubstance*, *DNA*, *RNA*, *Amplifications*, *FluorescentProtein*, *Oligonucleotide* and *RestrictionEnzyme*. Each model provides a rich set of features which describes the various steps used in the preparation of these materials and how they are used in an experiment. The model also consists of *StandardOperatingProcedure* which describes the procedures and the protocols used in an *Experiment*. So in total, the OMERO database consists of 145 tables and the ReceptorLight Database consists of 35 tables.

6.4 CAESAR: Provenance Representation

To connect all the information used in an experiment and link this data with other datasets on the web, it is essential to express and integrate the heterogeneous data using semantic web technologies. To make use of semantic web technologies, the experimental data need to be machine-understandable. However, the experimental data and the image metadata are stored in relational databases in CAESAR. To semantically represent this data and at the same time avoid replication of data which is already stored in the relational database and the file repository, we use the ontology-based data access approach. The overall goal of this approach in CAESAR is to provide several high-quality services to the domain scientists without worrying about the underlying technologies.

In this section, we discuss how we create a graph of the provenance of the experiments on the domain of light microscopy imaging. In Chapter 4, we discussed the REPRODUCE-ME data model and the ontology generated from it. We use the REPRODUCE-ME ontology for the creation of this provenance graph.

Ontology-based data access (OBDA) is an approach developed in the mid-2000s to access the various data sources using ontologies [Poggi et al., 2008]. The ontologies

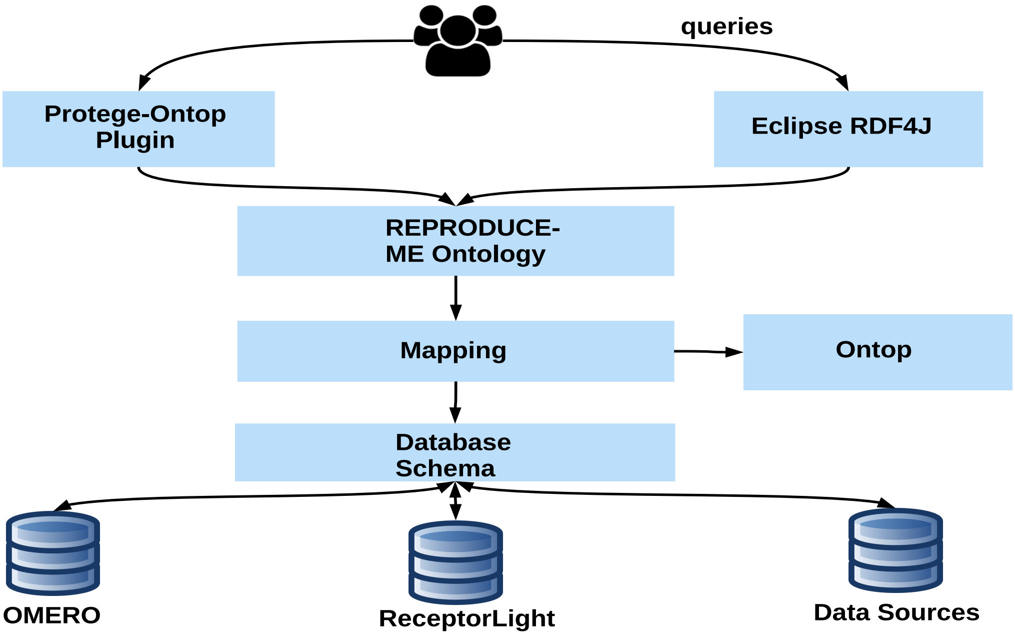


Figure 6.5: The OBDA in CAESAR

are used to represent the domain knowledge at an abstract level. In the OBDA approach, the details of the structure of the underlying data sources are isolated from the users using the high-level global schema provided by ontologies [Rodriguez-Muro et al., 2013]. Through this approach, the user is able to access the data source using the ontologies with the classes and properties. The relationship between the data and the terms in the ontologies are expressed using mappings. Thus the end-users do not need to know the technical details in the underlying relational databases including tables names, joins, etc. The data source is independent of the conceptual layer defined by the ontology. The user makes the queries based on the vocabulary using the classes and properties provided by the ontology. The queries made using the vocabulary are translated to SQL which queries the data in the database with the help of the mappings. The translation and the mappings are done using an ontology-based data access system. This approach helps to efficiently access a large amount of data from different sources and avoid replication of data which is already available in relational databases. Several applications that use OBDA have been widely used in different systems involving large data sources [Kharlamov et al., 2017, Brüggemann et al., 2016].

We use the ontology-based data access approach to access the various databases in CAESAR. The ontology-based data access system used in CAESAR is Ontop [Calvanese et al., 2017]. Figure 6.5 shows the OBDA in CAESAR. We explain each component in detail in the OBDA approach in CAESAR.

1 **Data Sources**

Currently, the ontology-based data approach map data from the OMERO and the ReceptorLight databases.

2 Ontologies

Ontologies are used to model the conceptual view of the world. In our case, the REPRODUCE-ME ontology is used to model scientific experiments. It provides the vocabulary of classes and properties to describe the complete path of a scientific experiment. The ontology also consists of classes and properties from OME data model[9] to describe the elements responsible for the image acquisition process in a microscope.

3 Federation

We use federation for the OMERO and ReceptorLight databases provided by the rdf4j SPARQL Endpoint.

4 Mappings

Mappings are the key features of the OBDA approach. It shows the relationship between the ontology terms and the relational schema. The relational data in the OMERO and the ReceptorLight database is mapped to the REPRODUCE-ME ontology using the OBDA approach using Ontop. Declarative mappings are used in Ontop to encode the relational data to the ontological terms. Ontop provides two ways to specify mappings: 1) W3C standard R2RML mapping language [Das et al., 2012]. 2) Ontop's native mapping language. We use Ontop's native mapping language which is easy to learn and use. Ontop also provides the user the facility to convert the mappings into R2RML mappings and vice-versa. The native mapping language provided by Ontop consists of a source and a target. The source is an SQL query which retrieves data from the database. The target defines RDF triples with the values provided by the source. The mappings were written using the *Protege* plugin provided by Ontop which is based on OWL API. Using the graphical interface provided by the plugin, it is able to create and edit mappings and execute SPARQL queries. The plugin which bootstraps the ontology and the mappings from the database plays a key role in developing the mappings.

We show some mappings where we integrate experiment with the settings of the instruments used in them using the REPRODUCE-ME ontology. Listing 6.1 shows a part of the mapping for the Experiment with its research context.

[9] `http://www.openmicroscopy.org/Schemas/Documentation/Generated/OME-2016-06/ome_xsd.html`

```
target
:Experiment_{uid} a :Experiment;
  prov:startedAtTime {date}^^xsd:dateTime;
  prov:generatedAtTime {creationdatetime}^^xsd:dateTime;
  :name {name} ; :id {uid}^^xsd:integer;
  :status {status}^^xsd:integer; :description {description};
  prov:wasDerivedFrom :Experiment_{originalobjectid};
  prov:wasAttributedTo :Researchgroup_Experiment_{uid}, :
   ContactPerson_Experiment_{uid}, :Project_Experiment_{uid
   }, :ExperimenterGroup_{ownergroupid};
  :hasDataset :dataset_{datasetid}; rdfs:label "Experiment
   "^^xsd:string .
source
select * from experimenttable
```

Listing 6.1: Mapping for Experiment

Listing 6.2 shows a part of the mapping for the Dataset and the images that it contains.

```
target
:dataset_{parent} prov:hadMember :image_{child} .
source
SELECT * FROM "datasetimagelink"
```

Listing 6.2: Mapping for Dataset and Image

Listing 6.3 shows a part of the mapping for the Experiment and the corresponding images.

```
target
:experiment_{experiment_id} p-plan:correspondsToVariable :
   image_{image_id} .
source
SELECT "image"."id" AS image_id, "experiment"."id" AS
   experiment_id FROM "image", "experiment" WHERE "image"."
   experiment" = "experiment"."id"
```

Listing 6.3: Mapping for Experiment and the corresponding images

Listing 6.4 shows a part of the mapping how the instruments are related to images.

```
target
:instrument_{instrument_id} p-plan:correspondsToVariable :
   image_{image_id} .
source
SELECT "image"."id" AS image_id, "instrument"."id" AS
   instrument_id FROM "image", "instrument" WHERE "image"."
   instrument" = "instrument"."id"
```

Listing 6.4: Mapping for Instrument and Image

Listing 6.5 shows a part of the mapping for the Microscope used in the generation of images.

```
target
:microscope_{id} a :Microscope ;
  :id {id}^^xsd:integer ;
  :hasSetting :lotnumber_microscope_{id} , :
   model_microscope_{id} , :serialnumber_microscope_{id};
   prov:wasAttributedTo :manufacturer_microscope_{id} ;
  :version {version}^^xsd:integer ; rdfs:label "Microscope
   "^^xsd:string .
source
SELECT * FROM "microscope"
```

Listing 6.5: Mapping for Microscope

Listing 6.6 shows a part of the mapping for the Instruments and their components.

```
target
:microscope_{microscope_id} :isPartOf :instrument_{
   instrument_id} .
source
SELECT "instrument"."id" AS instrument_id, "microscope"."id"
    AS microscope_id FROM "instrument", "microscope" WHERE "
   instrument"."microscope" = "microscope"."id"
```

Listing 6.6: Mapping for Microscope and Instrument

Listing 6.7 shows a part of the mapping for the different types of microscope.

```
target
:microscopetype_{microscopetype_id} prov:specializationOf :
   microscope_{microscope_id} .
source
SELECT "microscope"."id" AS microscope_id, "microscopetype
   "."id" AS microscopetype_id FROM "microscope", "
   microscopetype" WHERE "microscope"."type" = "
   microscopetype"."id"
```

Listing 6.7: Mapping for Microscope Types

Listing 6.8 shows a part of the mapping for the Objective, an Instrument component.

```
target
:objective_{id} a :Objective ; :id {id}^^xsd:integer ;
:hasSetting :calibratedmagnification_objective_{id} , :
   iris_objective_{id} , :lensna_objective_{id} , :
   lotnumber_objective_{id} , :manufacturer_objective_{id} ,
    :model_objective_{id} , :nominalmagnification_objective_
   {id} , :serialnumber_objective_{id} , :
   workingdistance_objective_{id} ; :version {version}^^
   xsd:integer ; rdfs:label "Objective"^^xsd:string .
source
SELECT * FROM "objective"
```

Listing 6.8: Mapping for Objective

Listing 6.9 shows a part of the mapping for the Image with the Objective that is associated with it.

```
target
:image_{image_id} :hasSetting :objectivesettings_{
   objectivesettings_id} .
source
SELECT "image"."id" AS image_id, "objectivesettings"."id" AS
    objectivesettings_id FROM "image", "objectivesettings"
   WHERE "image"."objectivesettings" = "objectivesettings"."
   id"
```

Listing 6.9: Mapping for Objective and Image

Listing 6.10 shows a part of the mapping for the Objective and its settings.

```
target
:objective_{objective_id} :hasSetting :objectivesettings_{
   objectivesettings_id} .
source
SELECT "objectivesettings"."id" AS objectivesettings_id, "
   objective"."id" AS objective_id FROM "objectivesettings",
    "objective" WHERE "objectivesettings"."objective" = "
   objective"."id"
```

Listing 6.10: Mapping for Objective and its settings

The mapping in Listing 6.1 shows how an experiment and its attributes are mapped to the ontology. Each experiment has one dataset which contains images (see Listing 6.2). The mapping in Listing 6.4 presents the relationship between the instrument and the image asscoiated with it. The following mappings shows the different parts of an instrument and their types and the settings. There are around 800 mappings to create the virtual RDF graph. All the mappings are publicly available[10].

[10] https://sheeba-samuel.github.io/REPRODUCE-ME/resources.html

5 **Query answering in OBDA**
A virtual RDF graph is created in OBDA using the ontology with the mappings [Calvanese et al., 2017]. This graph is queried using SPARQL, which is the standard query language in the semantic web community. The RDF graphs generated in this way can either be materialized or kept as it is as virtual. When the RDF graphs are materialized, RDF triples are generated which can be directly used in the RDF triplestores. The RDF graph can be kept virtual and queried when needed. We used the latter approach where the RDF graphs are kept virtual and queried only during query execution. The queries are executed in the visualization module mentioned in Section 6.7. The virtual approach helps to avoid the materialization cost and provides the benefits of the matured relational database systems.

Advantages The advantages of using this approach in CAESAR are:

- A virtual approach to have a view-based query answering without moving the data from the databases to the views or data warehouse [Kharlamov et al., 2017].
- Easy to learn to use the mapping language.

Limitations
One of the challenges that we faced is the assumption that the user can formulate queries over ontologies [Kharlamov et al., 2017]. To overcome this challenge, we provide visualization features so that users can visualize the experiment. The two visualization modules try to answer the competency questions mentioned in Chapter 4. There are also limitations in the Ontop system due to unsupported functions and data types[11].

6.5 CAESAR: Computational Reproducibility

To support computational reproducibility and capture the complete path of a scientific experiment, we integrate ProvBook [Samuel and König-Ries, 2018b] with CAESAR. To use ProvBook in CAESAR, we also installed and integrated JupyterHub[12]. This helps to create a collaborative research environment for computational reproducibility. It provides a group of users access to computational notebooks without the need for additional installation and maintenance tasks. They provide a multi-user version of notebooks for the scientists using CAESAR. The notebooks are stored in the file repository of CAESAR. The scientists can create new computational notebooks, run and share them [Samuel and König-Ries, 2018a].

[11] https://github.com/ontop/ontop/wiki/ObdalibIssues
[12] https://jupyter.org/hub

Scientists can directly work with the images and other datasets linked to an experiment in CAESAR using Jupyter Notebooks. They can access the images stored along with the experiments using the API and perform processing or analysis on them. The processed images and datasets can be uploaded and linked to the original experiments to CAESAR using the APIs. The provenance of the execution of the notebooks is captured using ProvBook. The provenance difference feature provided by the ProvBook helps the users of CAESAR to compare the difference between two executions of the notebook (see Section 5.2.3).

CAESAR also provides the feature to link these Jupyter Notebooks to the step of an experiment that used them using the metadata editor (see Section 6.2). In this way, the experiment dataset in CAESAR contain both the non-computational and computational steps. The provenance of the notebook represented in RDF using ProvBook is linked to the experimental provenance graph. The Notebook is linked to the experiment in RDF using the object property *p-plan:isSubPlanOfPlan*. Hence, we create a knowledge graph of the provenance of experiments with their computational and non-computational steps.

CAESAR also fetches the metadata from the notebooks using the JupyterHub REST API. The fetched metadata includes the details of the notebook including the sessions and the users [Samuel and König-Ries, 2018a]. The experimental data provided by scientists through the metadata editor, the metadata extracted from the images, and the details of the computational steps collected together are integrated, linked and represented using the REPRODUCE-ME ontology. All this provenance data together form the basis for the complete path of a scientific experiment.

6.6 CAESAR: Provenance Query and Difference

We showed how we captured, represented and stored provenance of experiments in CAESAR. An end-to-end provenance management solution should provide possibilities to query this data. In CAESAR, the data is stored in relational databases. It provides a way to query this information in SQL. However, querying data through heterogeneous data sources and linking resources from web is better possible through the usage of semantic web technologies. So we used the ontology-based data approach to represent this information in a machine-understandable format. The virtual graph created through this approach can be queried using SPARQL. For advanced users, the system provides a SPARQL editor to query the semantic data from the rdf4j-workbench. The users can write their own SPARQL queries to get the answer for the competency questions like **CQ10**. We provide some example SPARQL templates to help users to formulate queries. The SPARQL queries for the competency questions are provided in Section 7.5. In addition to that, there are APIs in CAESAR which can be used to get information on the experiments, materials and other datasets. These APIs can be directly used in scripts or Jupyter Notebooks to access the data and images. CAESAR provides the feature to compare different executions of computational steps using ProvBook. In a collaborative environment provided by CAESAR, it is important to know who created, modified, or executed the notebooks. Therefore, the changes made by different users need to be tracked and compared. Hence, ProvBook provides additional benefit in such scenarios. In addition to that, we also provide a basic comparison of different versions of experiment descriptions in CAESAR. This comparison shows the version history of the creation and modification of the data by different users in a group.

6.7 CAESAR: Provenance Visualization

A large amount of complex and heterogeneous data are generated in many life-science experiments. We have shown how we capture the experimental data and interlink with the concepts in the web using linked data in the previous sections. The captured experimental data needs to be visualized in a way that will help scientist for a better understanding of experimental processes and the dependencies. Data visualization is an important step in the provenance lifecycle. It is an effective and efficient medium of visual communication for any type of users. It helps them to understand the data and the factors that led to the final result. Scientific workflow management systems provide visualization of only the intermediate and final results along with the computational steps. But to get the complete picture of an experiment, the experimental metadata, the processes and the configurations of both the computational and non-computational steps are desirable. This is missing in such systems. It is important that the scientists visualize the data and the results along with its provenance information.

Scientific workflow management systems provide the traditional node-link visualization of workflows. Vistrails [Callahan et al., 2006] provides users the ability to compare different versions of workflows and their results. The InProv [Borkin et al., 2013] tool visualizes provenance of filesystem using radial layout. However, the focus of visualization is on the relationship between files and processes and the interactions between them. The visual encoding provided by the radial layouts requires right grouping method to effectively visualize the provenance data. However, to visualize the path, the traditional node-link diagram is more effective.

Provenance Map Orbiter [Macko and Seltzer, 2011] is another visualization tool which provides an exploration of large provenance graphs using graph summarizations and semantic zoom. Semantic zoom allows the user to drill down each node by zooming into them. This helps the user to visualize larger provenance graphs. Graph summarization uses summarization algorithm by creating summary nodes by combining objects of similar types. But in the cases where provenance graph does not contain enough semantic information to generate summarization will result in a large number of summary nodes thus resulting in large provenance graphs.

PROV-O-Viz [Hoekstra and Groth, 2015] is a web-based provenance visualization tool which is compatible with the PROV model. This tool uses Sankey Diagrams to visualize provenance traces focusing on activities. It visualizes the important activities and the data flow within a selected activity. The provenance graph is visualized by pasting the PROV-O text or connecting to a SPARQL endpoint. In our approach, we focus not only on the activity nodes but also on the other nodes from PROV-O and P-Plan.

The paper [Kunde et al., 2008] presents the abstract types of user requirements for a provenance visualization component. They are Process, Results, Relationships, Timeline, Participation, Compare, and Interpretation. Data visualization can be categorized in two ways: Exploration and explanation [Steele and Iliinsky, 2011]. Exploratory data visualizations are required in situations when scientists have a large amount of data and unaware of what data is in it. The datasets are visualized to tell the story the data has to offer. These visualizations are used in the data analysis phase. Explanatory data visualizations are used when the scientists already know about the data and need to tell its story to other scientists. This visualization is used in the presentation phase. There is another category which combines these two visualizations together: Hybrid. This visualization presents the data with the aim to allow exploration from the reader's part.

Our goal of the research is to provide data visualization which helps in the storytelling of an experiment. In this section, we present our visualization modules which are helpful for the data analysis phase and data presentation phase. Using Explanatory data visualization, we aim to selectively provide information so that the reader will able to understand it and receive the message. We follow the methodology proposed by [Steele and Iliinsky, 2011, Lee et al., 2006] in designing a visualization component. The first step in designing a visualization component is to understand the goal. We present our two goals in designing the visualization component in CAESAR:

- Provide users with a complete picture of an experiment
- Provide users the ability to track the provenance of an experiment

The next step of data visualization component design is to understand the dimensions of data that need to be communicated to the user. In order to do that, we frame questions that the visualization component needs to address. These questions are based on the competency questions mentioned in Section 4.2.

- Which are the entities that need to be communicated for the visualization of the complete path of a scientific experiment?

- What are the key relationships that are relevant to track the provenance?
- What is the complete picture of a scientific experiment?
- Which are the values and properties that are needed to track the provenance path?

Lee et al. [Lee et al., 2006] presents a list of tasks for the design of graph visualization systems which includes Topology-based tasks, Attributes-based tasks, Browsing tasks, Overview Tasks, and High-level tasks. Stitz et al. [Stitz et al., 2016] refined Lee's et al. tasks to support data visualization provenance graph in the Refinery platform. Based on these works, we define the features that we aim to address in designing the data visualization component in CAESAR.

1. Complete overview
2. User interaction
3. Interoperable
4. Interlinking of data
5. Comparison of experimental data and executions
6. Nested Hierarchy
7. View the node-link details

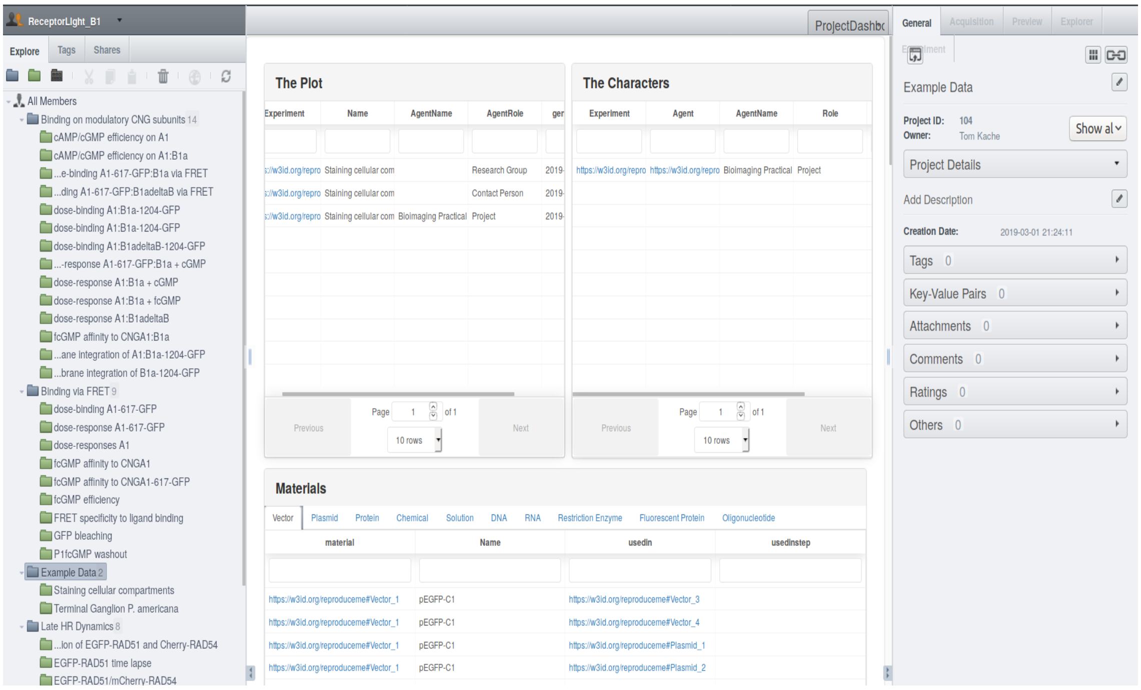
The experimental data captured and managed in CAESAR needs to be efficiently visualized for the scientists. We explained how we capture and represent the complete path of a scientific experiment consisting of computational and non-computational steps in Section 6.5. This module provides the users with an overview of an experiment by visualizing the complete path of an experiment. The module provides two components for viewing and accessing the provenance data:

- Dashboard
- ProvTrack

6.7.1 Dashboard

The Dashboard aggregates all the data related to an experiment at a single place to view. CAESAR offers a dashboard at the project and experiment level. The Project Dashboard is activated when a project is selected by the user while the experiment dashboard is activated when a dataset is selected. Each dataset has at most one experiment.

The Project Dashboard aggregates the provenance data from all the experiments


ReceptorLight_B1
Explore
Tags
Shares
All Members
Binding on modulatory CNG subunits 14
cAMP/cGMP efficiency on A1
cAMP/cGMP efficiency on A1:B1a
...e-binding A1-617-GFP:B1a via FRET
...ding A1-617-GFP:B1adeltaB via FRET
dose-binding A1:B1a-1204-GFP
dose-binding A1:B1a-1204-GFP
dose-binding A1:B1adeltaB-1204-GFP
...-response A1-617-GFP:B1a + cGMP
dose-response A1:B1a + cGMP
dose-response A1:B1a + fcGMP
dose-response A1:B1adeltaB
fcGMP affinity to CNGA1:B1a
...ane integration of A1:B1a-1204-GFP
...brane integration of B1a-1204-GFP
Binding via FRET 9
dose-binding A1-617-GFP
dose-response A1-617-GFP
dose-responses A1
fcGMP affinity to CNGA1
fcGMP affinity to CNGA1-617-GFP
fcGMP efficiency
FRET specificity to ligand binding
GFP bleaching
P1fcGMP washout
Example Data 2
Staining cellular compartments
Terminal Ganglion P. americana
Late HR Dynamics 8
...ion of EGFP-RAD51 and Cherry-RAD54
EGFP-RAD51 time lapse
EGFP-RAD51/mCherry-RAD54
ProjectDashbc
General
Acquisition
Preview
Explorer
Experiment
Previous
Page 1 of 1
10 rows
Next
Vector
Plasmid
Protein
Chemical
Solution
DNA
RNA
Restriction Enzyme
Fluorescent Protein
Oligonucleotide
Show al
Project Details
Add Description
Tags 0
Key-Value Pairs 0
Attachments 0
Comments 0
Ratings 0
Others 0

The Plot

Experiment	Name	AgentName	AgentRole	ger
s://w3id.org/repro	Staining cellular com		Research Group	2019-
s://w3id.org/repro	Staining cellular com		Contact Person	2019-
s://w3id.org/repro	Staining cellular com	Bioimaging Practical	Project	2019-

The Characters

Experiment	Agent	AgentName	Role
https://w3id.org/repro	https://w3id.org/repro	Bioimaging Practical	Project

Materials

material	Name	usedin	usedinstep
https://w3id.org/reproduceme#Vector_1	pEGFP-C1	https://w3id.org/reproduceme#Vector_3	
https://w3id.org/reproduceme#Vector_1	pEGFP-C1	https://w3id.org/reproduceme#Vector_4	
https://w3id.org/reproduceme#Vector_1	pEGFP-C1	https://w3id.org/reproduceme#Plasmid_1	
https://w3id.org/reproduceme#Vector_1	pEGFP-C1	https://w3id.org/reproduceme#Plasmid_2	

Example Data

Project ID: 104
Owner: Tom Kache

Creation Date: 2019-03-01 21:24:11

Figure 6.6: The Project Dashboard in CAESAR for the complete overview of experiments conducted in a project

in a project. Figure 6.6 shows a part of the project dashboard. The dashboard consists of several panels. Each panel serves a purpose giving a detailed view of an experiment. The data inside a panel is represented in a tabular manner. The panels are arranged in a way that it tells the story of an experiment [Samuel et al., 2018]. The following components form a story: plot, characters, background context, settings, events, conflicts, climax, and the final message. To understand the climax and message of the story, it is important to know the characters, the background context and the flow of the story. Similarly, to understand a scientific experimental result, it is important to know the agents, execution environment, and the workflow. The CAESAR provides a dashboard that tells the story of an experiment. Each component of an experiment is defined as a panel. The panels are:

- *The Plot*
  This panel displays the plot of an experiment which includes Research Project, Research Group of the user the experiment belongs to and the date of execution of the experiment.

- *The Characters*
  The table includes all the agents that are involved in an experiment directly or indirectly. The table displays the name of the person, the name of the associated experiment, the step at which the agent was responsible and the role of the person in the experiment. The experimenter, principal investigator are some of the people who are directly involved in an experiment. While, the agents are responsible for the manufacturing of the materials used in the

experiment, the distributor of the sample is indirectly responsible for an experiment.

- *Materials*
  This panel shows all the materials that are used in an experiment. This table shows at which step of an experiment the materials were referenced. The properties of the materials are also available in the table.

- *External Resources*
  This panel shows the external resources that were referenced during the experiment lifecycle. The External Resources include the publications, files or other external annotations used in the experiment.

- *Files*
  This panel shows the files that were referenced during the experiment. It also includes the detail about the step at which these files were used.

- *Jupyter Notebooks*
  Experiments contain computational processes. These processes are executed either using scripts or computational notebooks. It shows the computational notebooks that are used in the experiment and the step at which they were used.

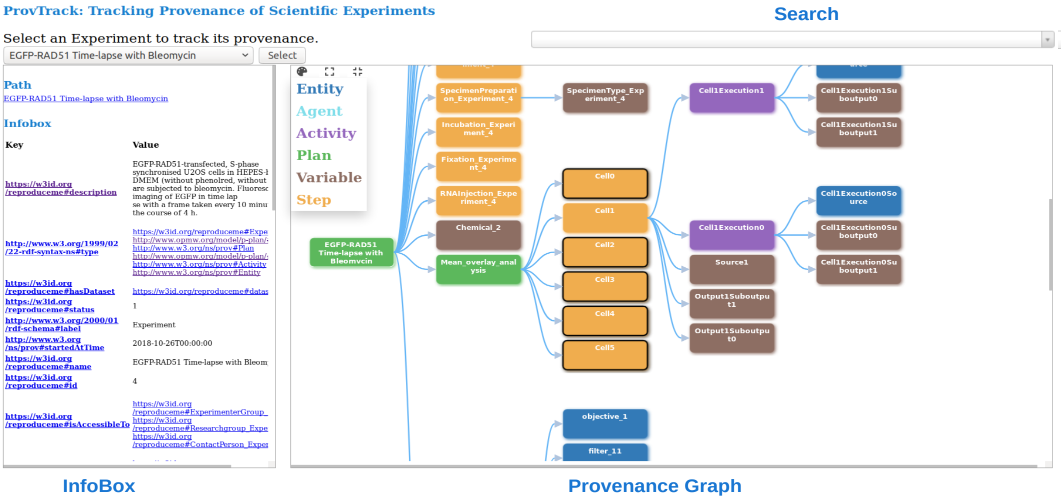
- *Steps/Activities*
  This provides a list of all steps and activities that are associated with an experiment.

- *Devices*
  The table shows all the devices used in the experiment along with their settings. This information is extracted from the images and the experimental data.

- *Settings*
  The table shows the settings of the devices used in the experiment. It includes the settings that were made during the experiment. These settings are extracted from the images and experimental data.

- *Results*
  This panel shows the results of a scientific experiment. It includes the final and intermediate results.

Each panel provides the answer to the competency questions **CQ1-CQ9** using SPARQL queries mentioned in Section 7.5. These panels also provide the user with the ability to search and filter the data based on keywords inside a table.



ProvTrack: Tracking Provenance of Scientific Experiments

Select an Experiment to track its provenance.

EGFP-RAD51 Time-lapse with Bleomycin ▾ Select

Path

EGFP-RAD51 Time-lapse with Bleomycin

Infobox

Key	Value
https://w3id.org/reproduceme#description	EGFP-RAD51-transfected, S-phase synchronised U2OS cells in HEPES-I DMEM (without phenolred, without are subjected to bleomycin. Fluoresc imaging of EGFP in time lap se with a frame taken every 10 minu the course of 4 h.
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	https://w3id.org/reproduceme#Expe http://www.opmw.org/model/p-plan/ http://www.w3.org/ns/prov#Plan http://www.opmw.org/model/p-plan/ http://www.w3.org/ns/prov#Activity http://www.w3.org/ns/prov#Entity
https://w3id.org/reproduceme#hasDataset	https://w3id.org/reproduceme#datas
https://w3id.org/reproduceme#status	1
http://www.w3.org/2000/01/rdf-schema#label	Experiment
http://www.w3.org/ns/prov#startedAtTime	2018-10-26T00:00:00
https://w3id.org/reproduceme#name	EGFP-RAD51 Time-lapse with Bleom
https://w3id.org/reproduceme#id	4
https://w3id.org/reproduceme#isAccessibleTo	https://w3id.org/reproduceme#ExperimenterGroup_ https://w3id.org/reproduceme#Researchgroup_Expe https://w3id.org/reproduceme#ContactPerson_Exper

Figure 6.7: ProvTrack: Tracking Provenance of Scientific Experiments

6.7.2 ProvTrack: Tracking the provenance of Scientific Experiments

Provenance data of scientific experiments can be complex and overwhelming. Representing such provenance data in tabular format may not be sufficient for the understanding of scientific experiments. To show the complete path of a scientific experiment, the graph representation with node and links is very helpful. In the dashboard, we provide an overview of all the experiments belonging to a project. In order to track the provenance of each scientific experiment, we present ProvTrack. ProvTrack is a visualization module to track the provenance of scientific experiments. Figure 6.7 shows the visualization of an experiment using ProvTrack. It provides users a visual and interactive way to track the provenance of results. It provides a node-link representation of provenance of experiments. Hence, it is possible for the user to backtrack the results. Also, it is possible to drill-down each node to get more information. The module is developed independently and integrated into CAESAR.

The interface provides the user with the options to select an experiment whose provenance needs to be tracked. When the user selects an experiment, the provenance graph is displayed. The user interface consists of three components. They are:

- Right panel

  The right panel provides an interactive provenance graph of an experiment. It consists of nodes and edges. Each node is colored based on its type. The type can be *prov:Entity*, *prov:Agent*, *prov:Activity*, *p-plan:Step*, *p-plan:Plan* and *p-plan:Variable*. The provenance graph is developed based on the REPRODUCE-ME data model. The *Expand All* button next to the help menu provides the facility to expand the provenance graph by opening up all the nodes. The *Collapse All* button collapses the provenance graph to just one node: *Experiment*. When a user hovers on an edge, it shows the property relationship between the two connecting nodes. The top left button in the right panel provides a help menu to highlight what each color means in the graph. Whenever a node is selected, the path from that node to the first node, which is *Experiment*, is highlighted. This helps the user to see where the node is in the provenance graph. Simultaneously, the path is also shown on top of the left panel.

- Left panel
  The Left panel provides additional information of the selected node in the right panel. It consists of *Infobox* of the selected experiment. Whenever a node is clicked, the information of the node is shown as a key-value pair. The key is either the object property or data property of the REPRODUCE-ME ontology which is associated with the selected node. The user can also click on each link to know what each property means. On top of the left panel, the path of the selected node is also shown. The path shows where the selected node is in the provenance graph. For example, when the *Experiment* node is selected, the *Infobox* provides the information on the agents, spatial and temporal properties, etc.

- Search
  The Search panel provides a dropdown to search for nodes and edges. Users can search for any entities in the graph defined by the REPRODUCE-ME data model. This is very helpful when the provenance graph is large.

The provenance graph shown by ProvTrack is based on the data model represented by REPRODUCE-ME ontology (Section 4.5). The SPARQL endpoint is queried to get the complete path of an experiment. To increase performance, several SPARQL queries are made and the results are combined together.

6.8 Implementation and Development

The system follows a Model-View-Controller architecture pattern for the development of CAESAR. The webclient is written in Python. Each module in the webclient uses Django-Python framework for its implementation. The Dashboard is implemented using ReactJs. The new services extended by OMERO.server is written in Java. The ProvTrack uses D3 JavaScript[13] library for the rendering of provenance graph.

[13] `https://d3js.org/`

6.9 Summary

This chapter presented the integration of all our research work together through CAESAR. CAESAR uses a provenance-based semantic approach for the understandability, reproducibility, and reuse of scientific experiments. It is developed on top of OMERO, which is an image-based data management platform. We discussed the architecture of OMERO and how it is extended to provide provenance management of experiments. The architecture of CAESAR consists of modules to capture, represent, store, query, compare and visualize the provenance of scientific experiments. Each module is important and plays a vital role in end-to-end provenance management of experiments. The modules are dependent on each other. The provenance capture of non-computational steps of scientific experiments is done using the Metadata editor provided in CAESAR. The provenance of computational steps of scientific experiments is captured using ProvBook integrated into CAESAR. The captured provenance is represented using the REPRODUCE-ME ontology using the ontology-based data access approach. Finally, the complete path of a scientific experiment is visualized using ProvTrack. It provides a graph-based representation with nodes representing each entity in the experiment and edges representing each property linking two entities. The dashboard, on the other hand, provides a complete overview of the experiments performed together for a project. The evaluation of CAESAR is presented in Chapter 7 and the results are provided in Section 7.5.

Chapter 7

Evaluation

In the previous chapters, we presented our three main contributions of this research work. In this chapter, we evaluate our work by using them in real-world scenarios. The evaluation of our work is done to validate the hypothesis 2.3 defined in Chapter 2.

7.1 Overview of the Evaluation

We evaluate different aspects of our research work based on our main hypothesis that it is possible to capture, represent, manage and visualize a complete path taken by a scientist in an experiment including the computational and non-computational steps to derive a path towards experimental results. We first focus on evaluating the hypothesis H1 and H2. We have shown in Chapter 4 how we have developed a data model using Semantic Web technologies to describe the complete path of a scientific experiment. We first evaluate our data model with the help of scientists from different disciplines. Section 7.2 presents the insights from the user-based interviews on the scientific data management for reproducibility. In Section 7.3, we present results from a user-survey conducted to understand experiments and research practices for reproducibility. These user-based surveys are conducted to evaluate whether the terms added in the REPRODUCE-ME ontology are required to describe the provenance of a scientific experiment. In the following Section 7.4, we evaluate the hypothesis H3 and H4 by performing data and user-based evaluation of ProvBook in different scenarios. In Section 7.5, we evaluate CAESAR based on the hypothesis H5. We use real-life experiments provided by scientists for answering the competency questions **CQ1-CQ24**. We evaluate the system based on the users' perspective and provide the results of the user study that we conducted in Section 7.5.2 to check whether the requirements R1-R7 are satisfied. We conclude the chapter with the summary of the evaluation results in Section 7.6.

7.2 User-based Interviews

It is important to comprehend the current research practices and the essential components required to understand and reproduce a scientific experiment. To evaluate our data model, we interviewed scientists from different domains to understand the prevailing research practices. Open oral interviews were conducted among scientists from fields like Biology, Chemistry, Biodiversity, and Ecology. A workshop on "Fostering reproducible science – What data management tools can do and should do for you" was conducted in conjunction with BEXIS2 UserDevConf[1] Conference. Around 40 researchers from the projects iDiv[2], Aquadiva[3], ReceptorLight[4], BEXIS2[5] and other scientists from Jena University participated in the workshop. This workshop was jointly organized by the data management teams of the BEXIS 2, iDiv, AquaDiva and ReceptorLight projects. The participants included PostDocs, PhD Students, Data Managers, and Research Staff. The aim of this workshop was to understand the current practices and the challenges the researchers are facing with regard to the reproducibility of scientific results. The participants were asked to answer the following questions:

1. How do you document your research process?
2. How do you ensure, you (and others) are able to find your data again in 5 years?
3. What do you do to make your data reusable for others?
4. How do you ensure that your research findings are reproducible by others?
5. What tools are you using to address the questions above?
6. What tools would help you to improve the preparation and management of your research data?

We present key points expressed by 10 researchers, who actively participated in the discussions, from their own experiences of their daily research work. The detailed points by these researchers are presented in Appendix A. These interviews and discussions helped us in the development and evaluation of the REPRODUCE-ME data model and CAESAR. The important points expressed by the researchers are summarized below:

[1] http://fusion.cs.uni-jena.de/bexis2userdevconf2017/workshop/

[2] https://www.idiv.de/

[3] http://www.aquadiva.uni-jena.de/

[4] http://www.receptorlight.uni-jena.de/

[5] http://bexis2.uni-jena.de/

1. Collaboration with researchers distributed geographically in bigger project consortiums throughout the research lifecycle from data creation to the publication of results is needed.

2. Version-controlled and citable results along with the experimental data and metadata are required.

3. Linking the experimental data, steps, and results are important for reproducibility.

4. Awareness of data management process is required.

5. There is a lack of resources for the proper management of data with concerns regarding space-constraints of public repositories.

6. Scripts are vital for data analysis. It is important to document what scripts are doing to understand the results.

7. Documentation of individual trials helps in understanding which possibilities did not work out.

7.3 Survey on Understanding Experiments and Research Practices for Reproducibility

A user-based evaluation was conducted using an online survey. The goal of this survey is two-fold. The general goal of this survey is to understand experiments and research practices for reproducibility in different domains. In addition to that, through this survey, we also evaluate whether the terms added in the REPRODUCE-ME Data Model are required for understanding and reproducing experiments.

7.3.1 Materials and Methods

We developed an online survey consisting of 26 questions grouped in 6 sections. The purpose of this study is to gain a better understanding of what is needed to achieve reproducibility of experiments in science. The six sections are *(1) Privacy policy*, *(2) Research context of the participant*, *(3) Reproducibility*, *(4) Measures to ensure reproducibility*, *(5) Important factors to understand a scientific experiment to enable reproducibility* and *(6) Experiment Workflow/Research Practices*. The survey questionnaire is available in Appendix C. The survey was completely anonymous. The average time taken by a participant to complete the survey was around 10 minutes. The survey was implemented using Limesurvey[6]. For compliance reasons,

[6] https://www.limesurvey.org/

we provided a privacy policy form to get consent of the participants before collecting any kind of personal information according to the General Data Protection Regulation (GDPR)[7] (in German: Datenschutz-Grundverordnung, DSGVO). None of the questions in the survey was mandatory apart from the privacy policy form. We provided 'Other' option with a facility to provide additional comments for a majority of the questions. We also provided 'Not applicable' option to some of the questions wherever applicable. The definitions of terms like 'Reproducibility', 'Reproducibility Crisis', 'Metadata', etc. were either provided on top of the sections or external links were given to their definitions. The survey was first validated by a group of four researchers from Computer Science and Biology before distributing to the participants.

Survey Method

The survey was made available online on 24th January 2019. The survey link was distributed to the scientists in the ReceptorLight project who are currently the direct users of the REPRODUCE-ME Data Model. It was also distributed to several departments in the University of Jena, Germany through internal mailing lists. Apart from the ReceptorLight project, it was also distributed among the members of the iDiv, BEXIS and AquaDiva projects. The members of the Michael Stifel Center Jena[8] which is a center to promote interdisciplinary research for Data-driven and Simulation Science also participated in this survey. It was also advertised using Twitter through the Fusion[9] group account[10]. It was also distributed through internal and public mailing lists including RDA-de(Research Data Alliance-Germany)[11] and JISCMail[12].

7.3.2 Survey Results

A total of **101** out of **150** respondents were considered eligible for the analysis of the results. The basic eligibility criteria include that the participants have read and agreed to the privacy policy. The participants who only filled their research context and skipped the rest were also excluded from the analysis. In the following sections, we present the analysis of each question from the survey. The survey results along with the raw data and graphs are available online [Samuel and König-Ries, 2019]. We present the discussion on the survey results in Section 7.3.3.

[7] https://dsgvo-gesetz.de/

[8] https://www.mscj.uni-jena.de/

[9] http://fusion.cs.uni-jena.de/

[10] https://twitter.com/fusionUniJena/status/1090544753635147776

[11] https://www.rda-deutschland.de/

[12] https://www.jiscmail.ac.uk/

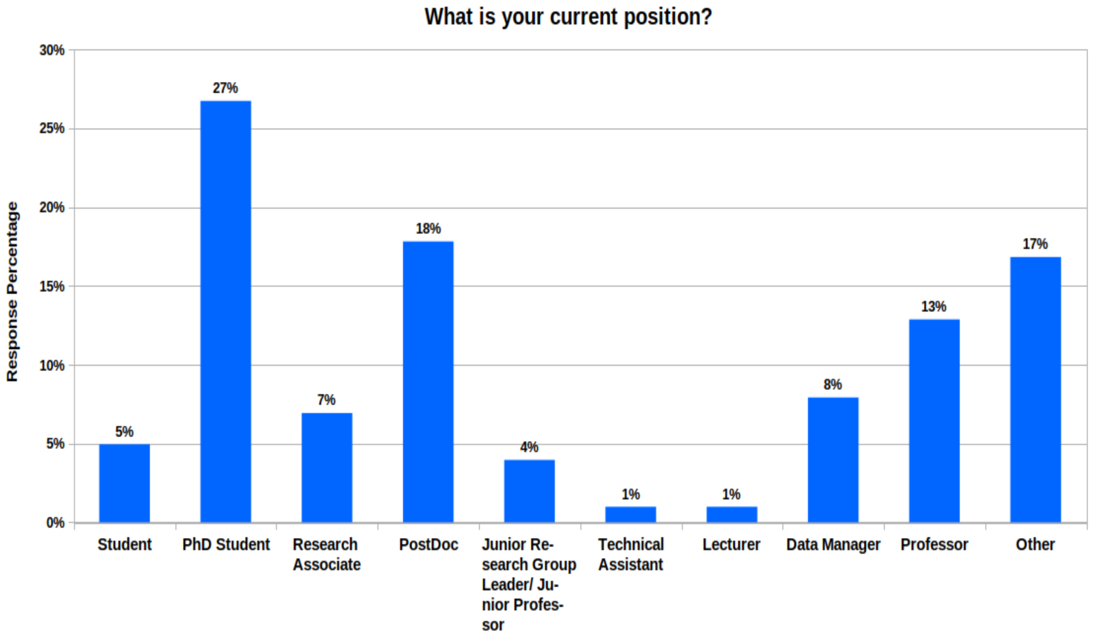


Figure 7.1: Current position of the survey participants

7.3.2.1 Research Background of the Participants

Figure 7.1 shows the current position held by the participants of the survey. Out of **101** respondents, **27%** of them were PhD students, **18%** of postdoctoral fellows, **5%** of Bachelor or Master students and **7%** of Research Associates. Around **17%** of the participants were either a Professor (**13%**) or Junior Research Leader/Professor (**4%**). The participants who selected "Other" include **6** librarians (**6%**), **3** software engineers (**3%**), **1** publisher and **7** other data officers.

The primary area of study of the participants is shown in Figure 7.2. The majority of the participants were from different fields of biology. They include molecular biology (**6%**), cell biology (**2%**), microbiology (**1%**) and biology (other) (**17%**). So in total, **26%** of participants come from different fields of biology. Participants from computer science (**19%**) and environmental sciences (**13%**) are other major contributors. The other participants come from fields like neuroscience (**6%**), chemistry (**1%**), plant sciences (**3%**), health sciences (**3%**) and physics (**4%**). The participants who selected the 'Other' option are **26%** and they come from various fields like biophysics, sociology, earth science, electrophysiology, engineering, etc.

7.3.2.2 Reproducibility Crisis and its causing factors

We asked the participants whether they think there is a reproducibility crisis or not. We had provided 3 options: *Yes*, *No* and *Other* with a free text field. **59%** of the participants think that there is a reproducibility crisis, while, **30%** of them think that there is no reproducibility crisis (Figure 7.3). **11%** of them selected the *Other* option and provided their views. **3** participants responded that there is partly crisis

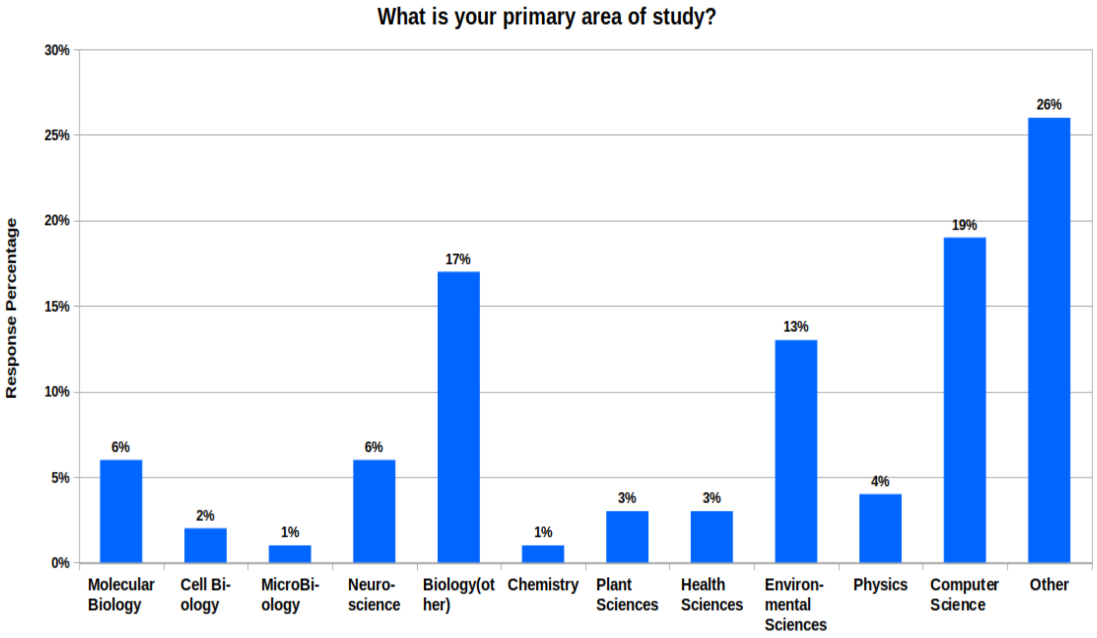


Figure 7.2: The primary area of study of the survey participants

while **3** others responded that they would not like to say the word 'crisis' instead pointed out there is a room for improvement and attention is required. The other comments include 'Depends on the scientific field', 'maybe', and 'I don't know'. Figure 7.4 and 7.5 show the view of the participants on reproducibility crisis analyzed based on their current position and primary area of study respectively. **74%** of the PhD Students and **72%** of PostDocs think that there is a reproducibility crisis (Figure 7.4). While, **54%** of professors do not believe that there is reproducibility crisis. **68%** of participants from computer science and **59%** from biology (other) believe in the existence of this crisis (Figure 7.5). **65%** of the participants coming from Molecular Biology, Cell Biology, Microbiology or Biology (other) think that there is a reproducibility crisis. The participants who either selected 'Yes' or 'Other' to this question were directed to the next question about the factors that lead to poor reproducibility from their own experiences. We provided 12 multiple-choice options including 'Other' with a free text field. As seen in Figure 7.6, **30%** of the total participants (**101**) who do not think there is a reproducibility crisis belong to the N/A (Not Applicable) category. The majority of the respondents consider that there is lack of data that is publicly available for use (**79%**), lack of sufficient metadata regarding the experiment (**75%**) and lack of complete information in the Methods/Standard Operating Procedures/Protocols (**73%**) as shown in Figure 7.7. The other reasons based on the majority votes include lack of time to follow reproducible research practices (**62%**), pressure to publish (**61%**), lack of knowledge or training on reproducible research practices (**59%**), lack of the information related to the settings used in original experiment (**52%**), poor experimental design (**37%**), data privacy (e.g. data sharing with third parties) (**34%**), Difficulty in understanding

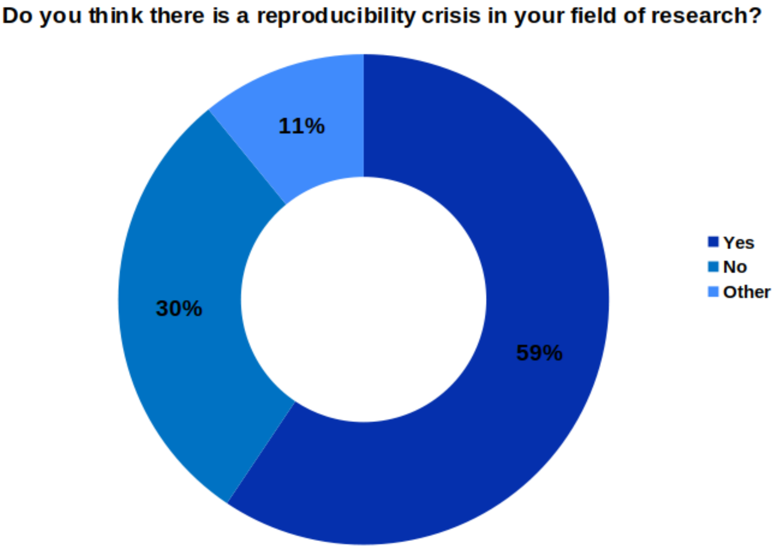


Figure 7.3: Do you think there is a reproducibility crisis in your field of research?

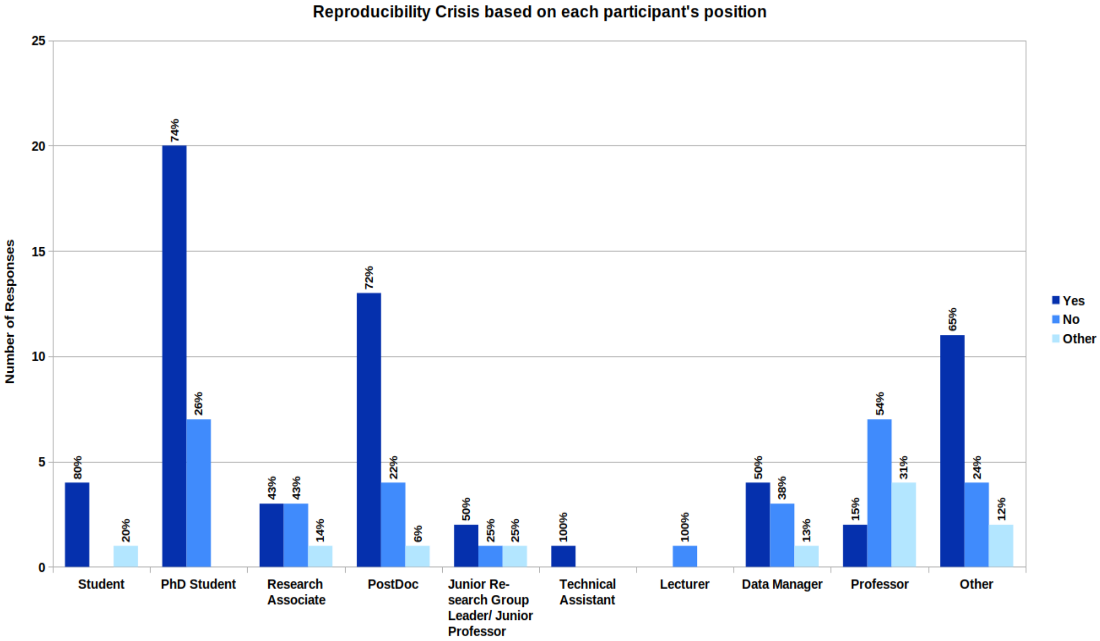


Figure 7.4: The view of participants on Reproducibility Crisis based on their position

laboratory notebook records (**20%**) and lack of resources like equipments/devices in workplace (**17%**).

7.3.2.3 Measures to ensure reproducibility

In the next section of the survey, we asked the participants about the measures taken in their field of research to ensure reproducibility. The first question was "How easy would it be for you to find all the experimental data related to your own project in order to reproduce the results at a later point in time (e.g. 6 months after the original experiment)?". We used 6-point scale for the answer options from *Very Easy* to *Very Difficult*. The subquestions included *Input Data*, *Metadata about*

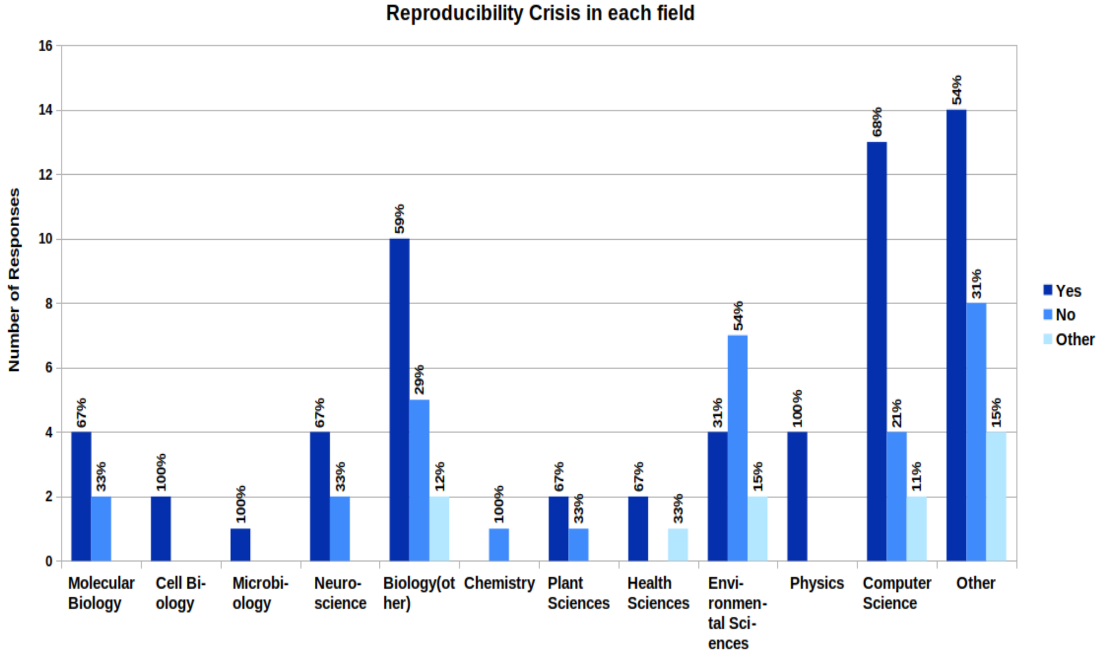


Figure 7.5: The view of participants on Reproducibility Crisis based on their area of study

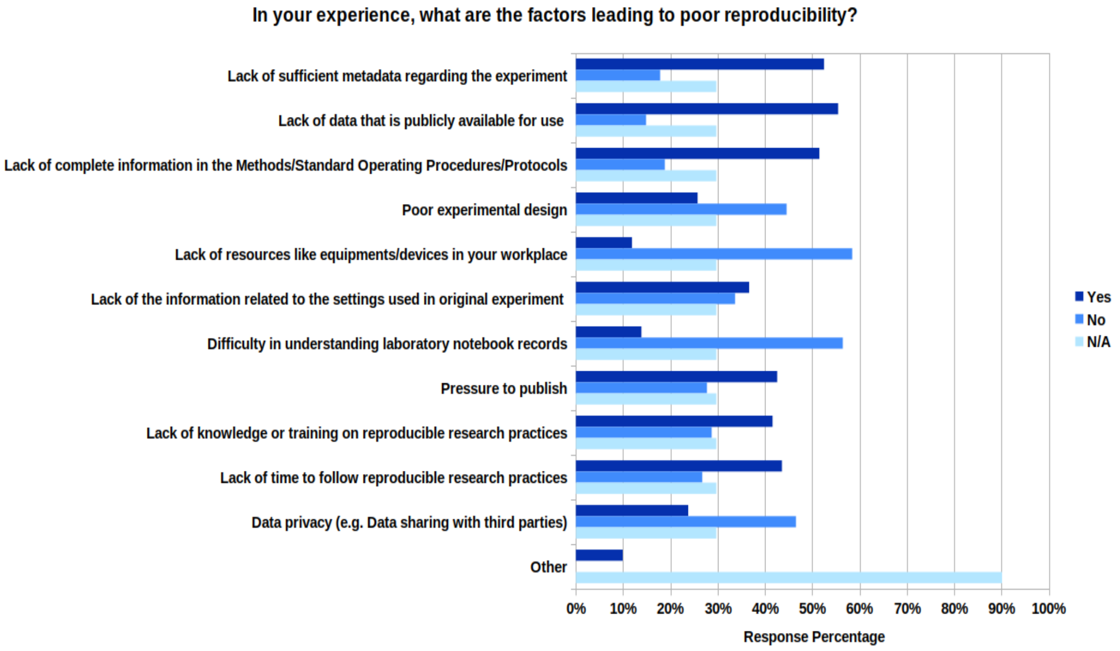


Figure 7.6: The factors leading to poor reproducibility from the experience of participants

the methods, *Metadata about the steps*, *Metadata about the experimental setup* and *Results*. **79%** of *Results* and **71%** of *Input Data* are either easy or very easy to find (Figure 7.8). But when it comes to the *Metadata about the steps* (**47%**) and *Metadata about the experimental setup* (**47%**), it gets less easy. The findability of *Metadata about the Steps* (**36%**), *setup* (**38%**), and *methods* (**32%**) shifts to neither easy nor difficult. According to the analysis, it is seen that the results and input data

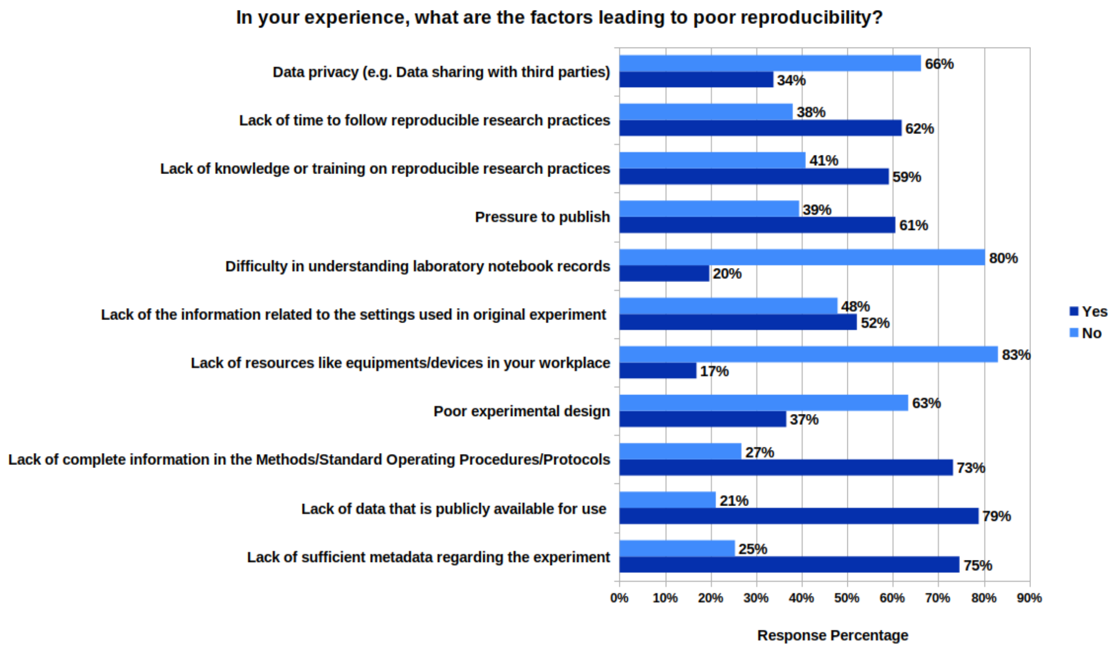


Figure 7.7: The factors leading to poor reproducibility from the experience of 71 participants who fully responded to this question.

are comparatively easier to find than the steps, methods and the setup metadata.

However, this trend changes when asked about a newcomer in their workplace to

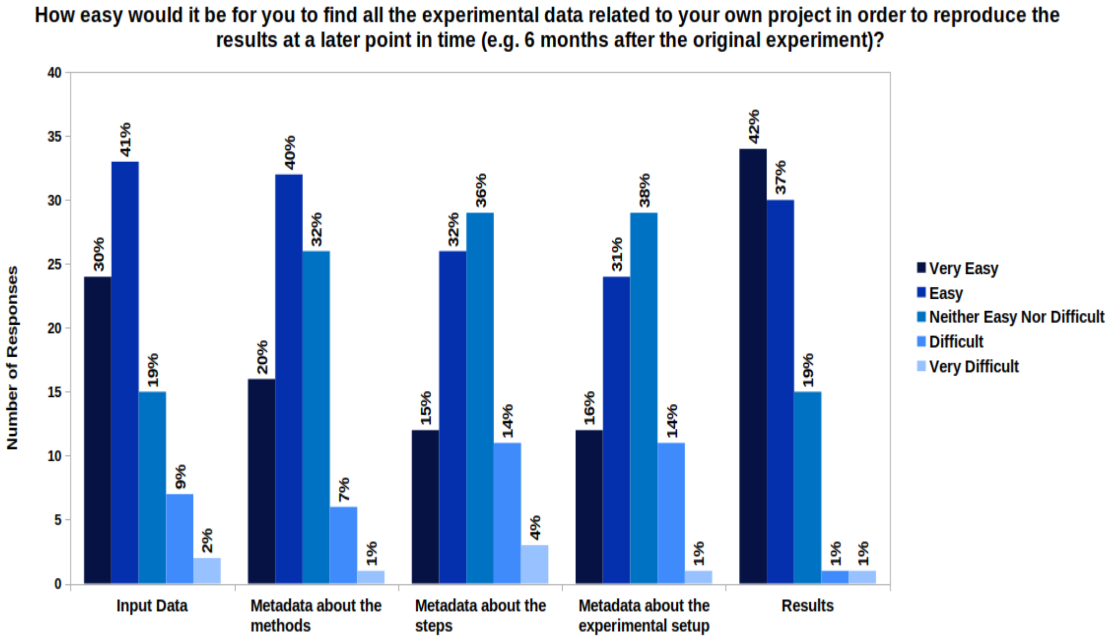


Figure 7.8: How easy would it be for you to find all the experimental data related to your own project in order to reproduce the results at a later point in time (e.g. 6 months after the original experiment)?

find the same experimental data of the participants without any/limited instructions from them (Figure 7.9). The percentage of easily finding the results and input data for a newcomer drops drastically from **79%** and **71%** to **48%** and **43%** respectively.

The most difficult metadata to find is about the steps (**48%**) and environment setup (**48%**). The difficulty to find the input data, methods, and results is **35%**, **35%** and **24%** respectively.

In the next question, they were asked whether they have ever been unable to reproduce published results of others. **54%** of them were unable to reproduce others published results, while **36%** of them said 'No' as seen in Figure 7.10. **10%** of them have never tried to reproduce others published results.

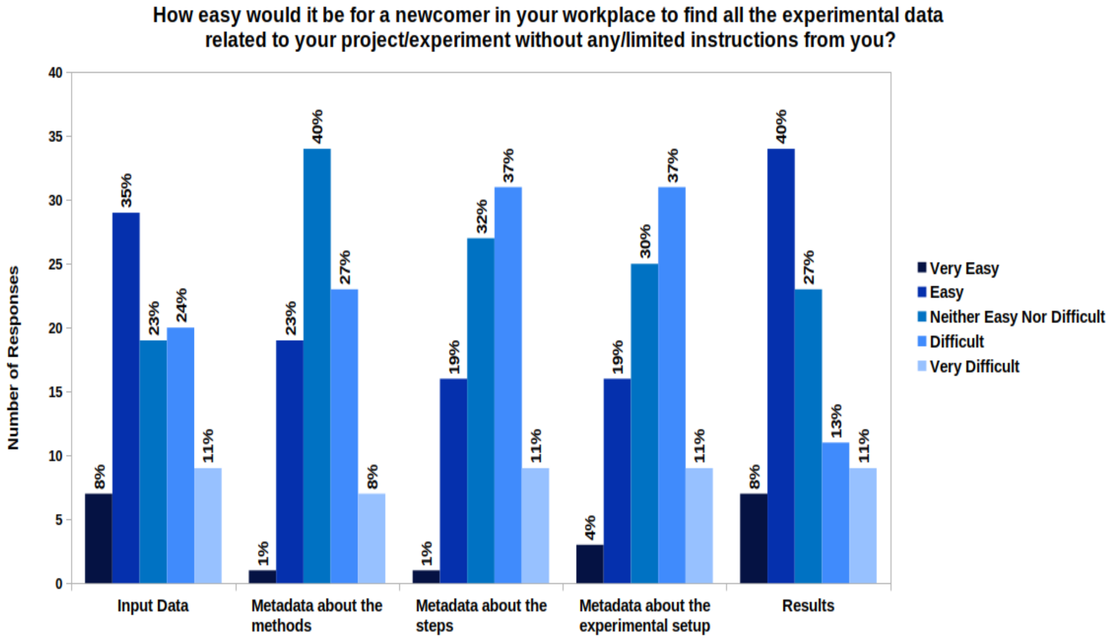


Figure 7.9: How easy would it be for a newcomer in your workplace to find all the experimental data related to your project/experiment without any/limited instructions from you?

The next question was "Has anybody contacted you that they have a problem in

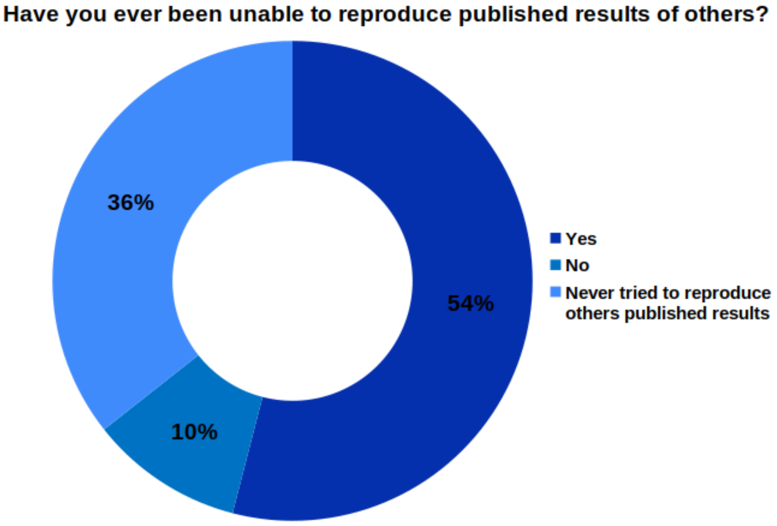


Figure 7.10: Have you ever been unable to reproduce published results of others?

reproducing your published results?". Even though we see through this survey that there exist issues regarding reproducibility, **95%** of the participants have never been contacted and only **5%** of them have been contacted concerning issues in reproducing their published results (Figure 7.11).

In the next question, "Do you repeat your experiments to verify the results?", **53%** of the respondents repeat their experiments, **12%** sometimes and **35%** of them do not repeat their experiments to verify their results.

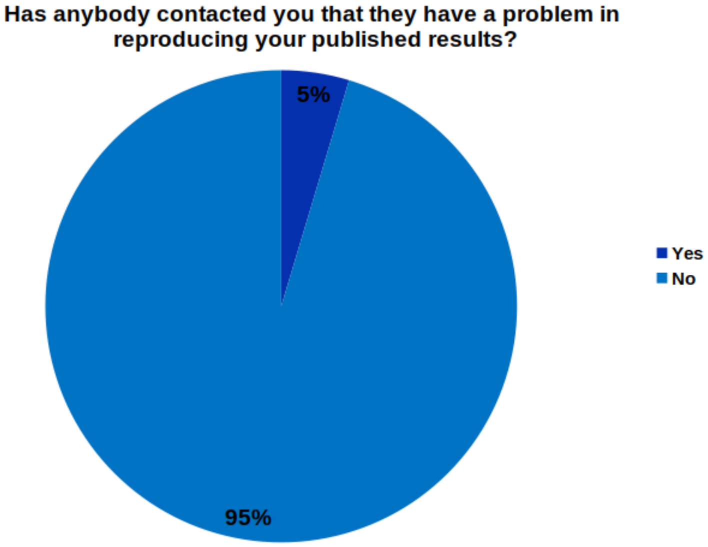


Figure 7.11: Has anybody contacted you that they have a problem in reproducing your published results?

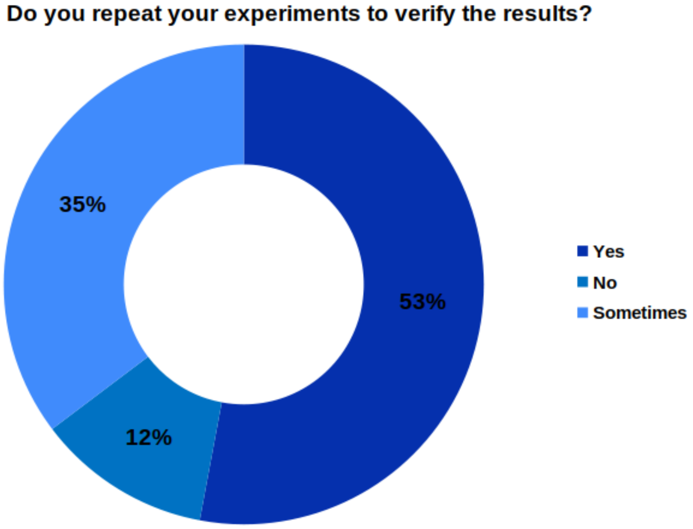


Figure 7.12: Do you repeat your experiments to verify the results?

7.3.2.4 Opinion on sharing experimental metadata

In this section of the survey, we asked about the factors that are important for them to understand a scientific experiment in their field of research to enable reproducibility. Here, we see what people think is important for the understandability and reproducibility of scientific experiments and see whether representing them in the REPRODUCE-ME data model is valid or not.

In the first question, we asked their opinion on sharing experimental data including Raw Data, Processed Data, Negative Results, Measurements, Scripts/Code/Program, Image Annotations, and Text Annotations. Surprisingly, **80%** of the participants shared their view that the negative results are either very important or absolutely essential while sharing data. As in the case for others, the participants consider sharing scripts (**78%**), processed data (**73%**), measurements (**71%**), raw data (**58%**), image annotations (**60%**) and text annotations (**55%**) either very important or absolutely essential.

84% of the participants consider that sharing the metadata about the experiment materials is either very important or absolutely essential while **81%** of them consider the same way for the instruments used in an experiment.

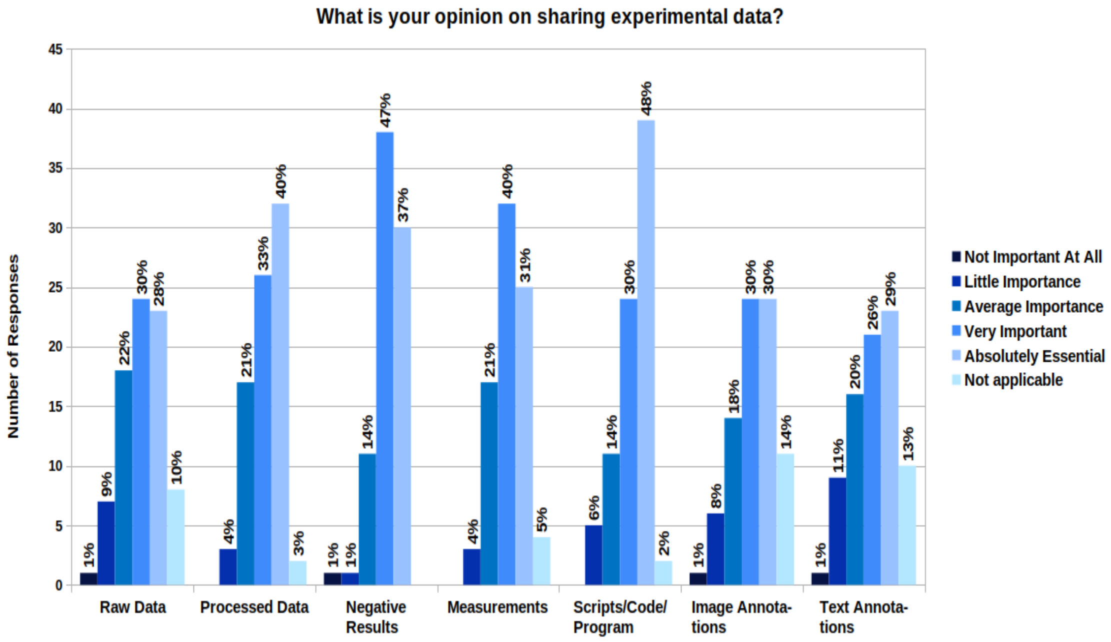


Figure 7.13: What is your opinion on sharing experimental data?

Participants consider that instrument settings (**80%**), experiment environment conditions (**76%**) and publications used (**68%**) are either very important or absolutely essential. Participants consider that it is very important or absolutely essential to share the names (**70%**), contacts (**65%**) and role (**54%**) of the agents who are directly involved in a scientific experiment. The participants also consider that the names (**20%**), contacts (**18%**) and role (**15%**) of the agents who are indirectly involved (like Manufacturer, Distributor) in a scientific experiment are very important or absolutely essential (see Figure 7.16). **50%** of the participants consider date as either very important or absolutely essential while **47%** of them consider the same way for time. **66%** of the participants consider duration as either very important or

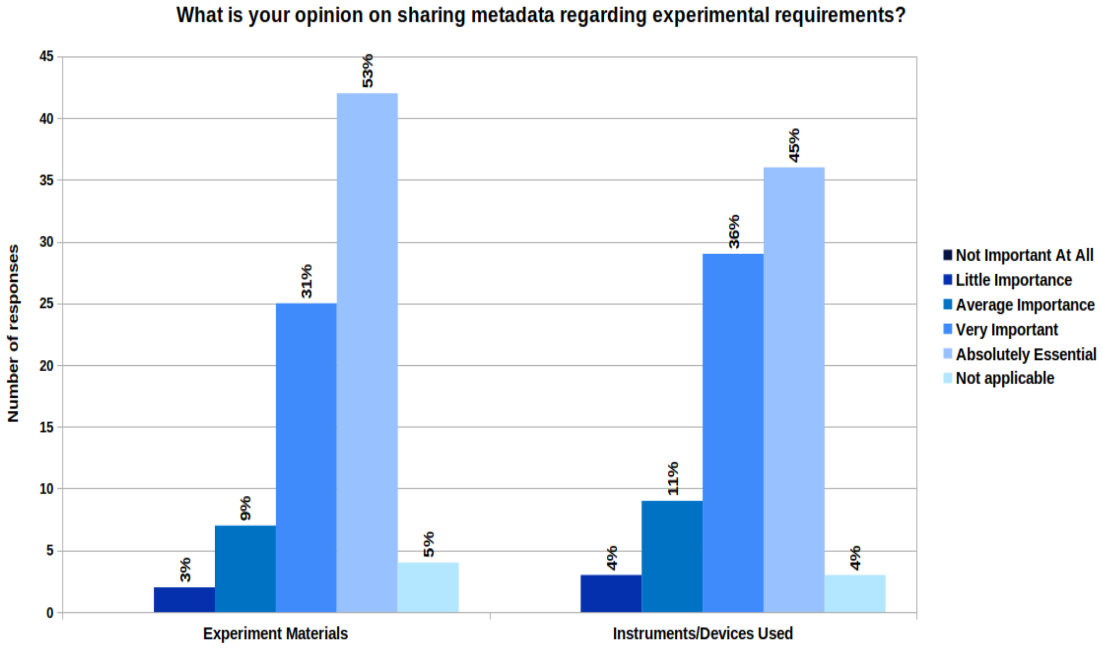


Figure 7.14: What is your opinion on sharing metadata regarding experimental requirements?

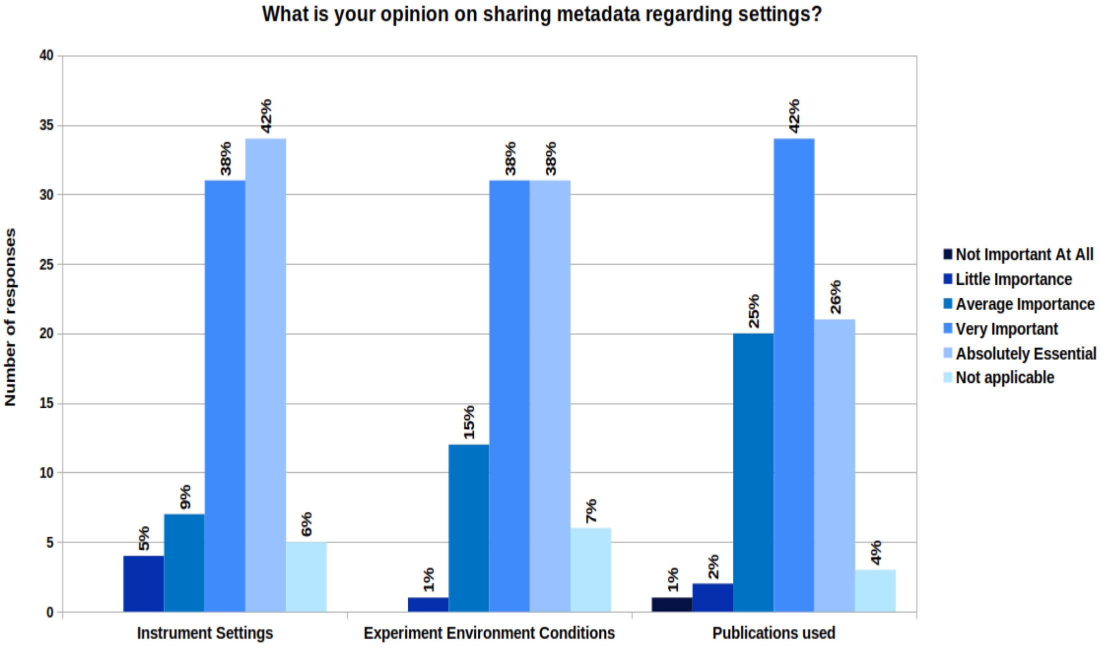


Figure 7.15: What is your opinion on sharing metadata regarding settings?

absolutely essential while **46%** of them consider the same way for location. Participants consider that software parameters (**80%**), software version (**77%**), software license (**37%**) and scripts/code/program used (**79%**) are either very important or absolutely essential. Participants also consider that Laboratory Protocols (**73%**), Methods (**93%**), Activities/Steps (**81%**), Order of Activities/Steps (**77%**), Validation Methods (**81%**) and Quality Control Methods used (**73%**) are either very important or absolutely essential. **86%** of the participants consider that final results

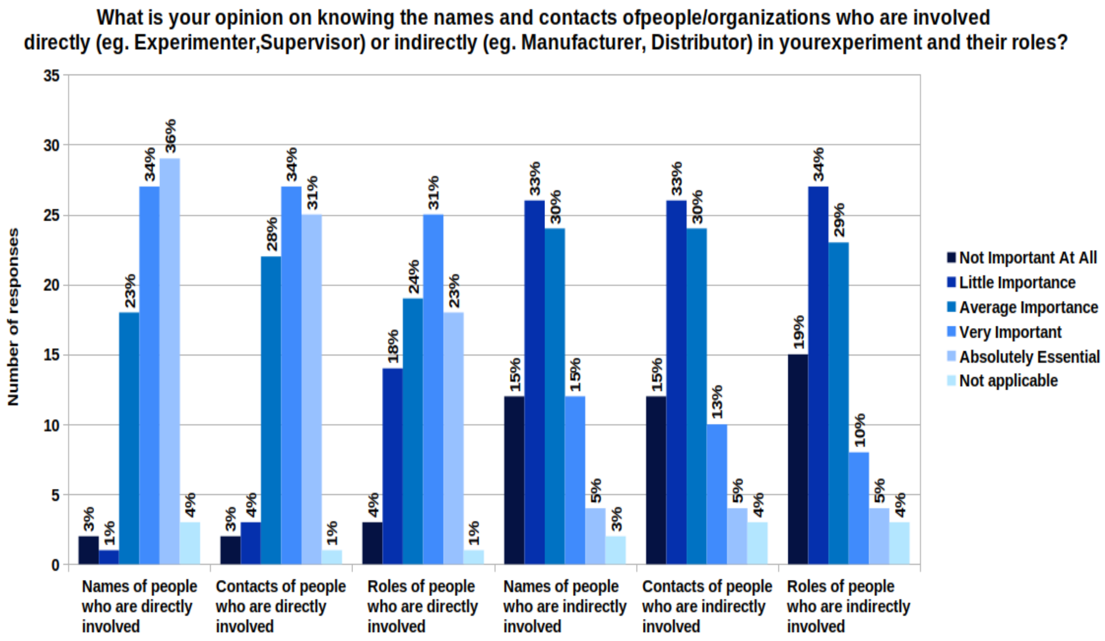


Figure 7.16: What is your opinion on knowing the names and contacts of people/organizations who are involved directly (eg. Experimenter, Supervisor) or indirectly (eg. Manufacturer, Distributor) in your experiment and their roles?

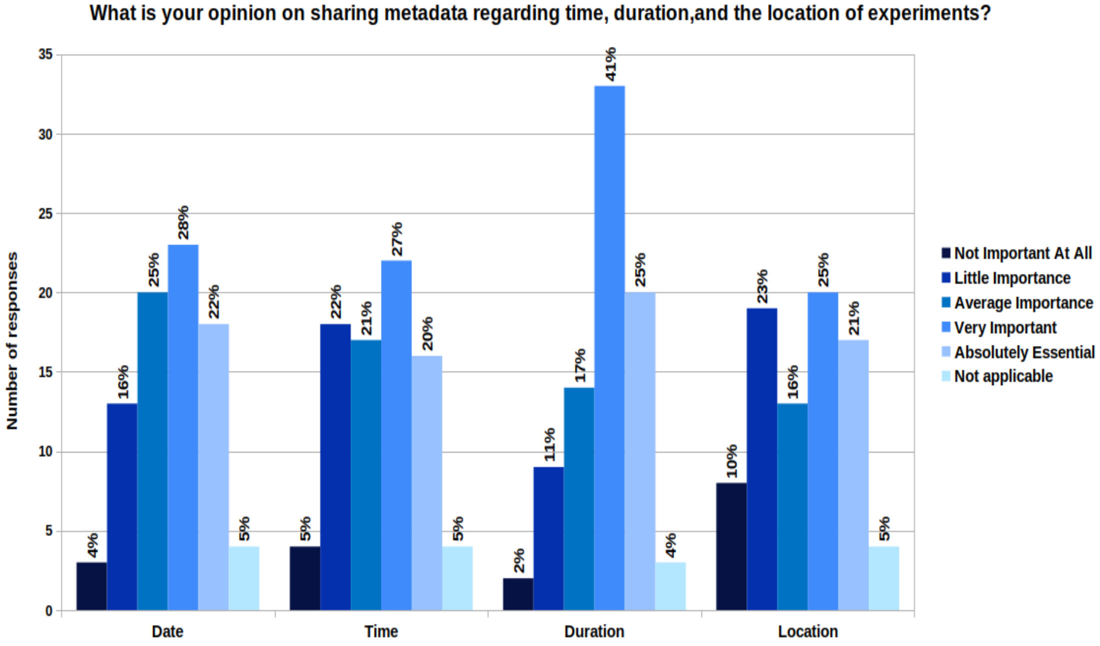


Figure 7.17: What is your opinion on sharing metadata regarding time, duration, and the location of experiments?

of each trial of an experiment are either very important or absolutely essential while **41%** of them think the same way for intermediate results. We had asked what else should be shared when publishing experimental results for which we got 12 responses which is provided in Appendix C.1.

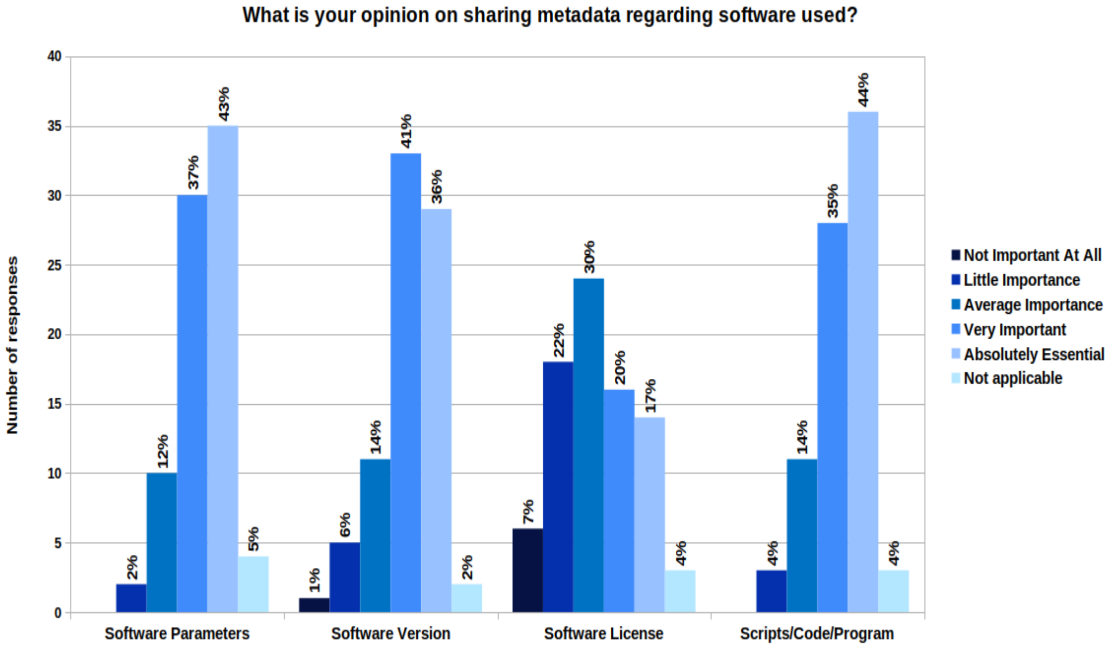


Figure 7.18: What is your opinion on sharing metadata regarding software used?

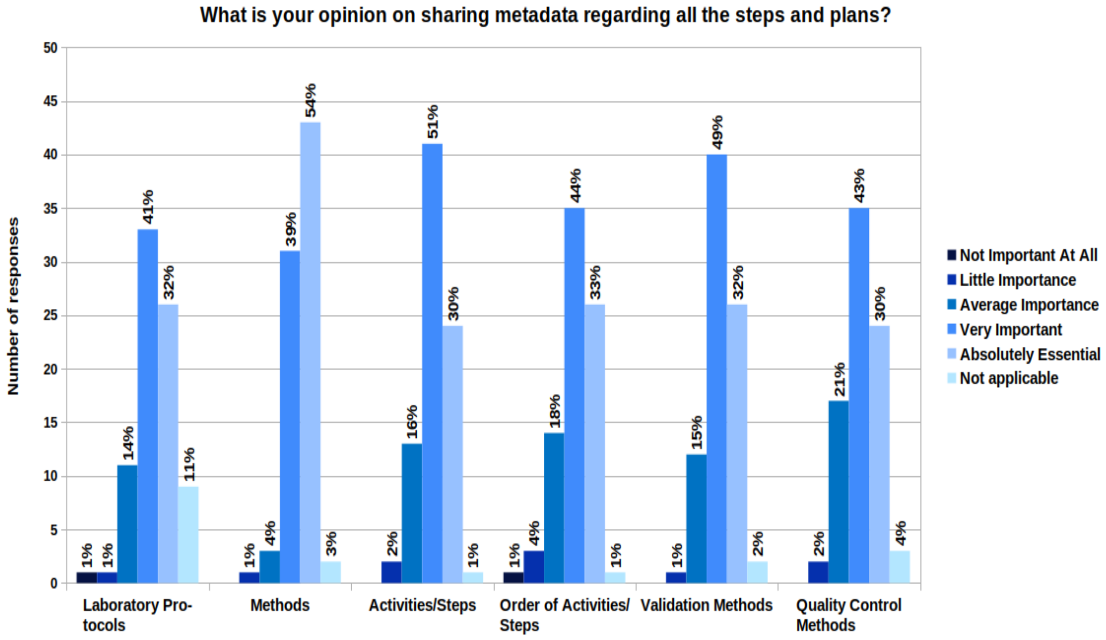


Figure 7.19: What is your opinion on sharing metadata regarding all the steps and plans?

7.3.2.5 Experiment Workflow/Research Practices

In this section, we asked about their experiment workflow and research practices. In the first question, we asked what kind of data they work primarily with. Figure 7.21 shows the distribution of the data they work with. Majority of them work with measurements (**27%**). The others work with images (**20%**), tabular data (**20%**), graphs (**20%**), and **8%** of them work with multimedia files. The participants who selected the 'Other' option work with text, code, molecular and geo-data. We next

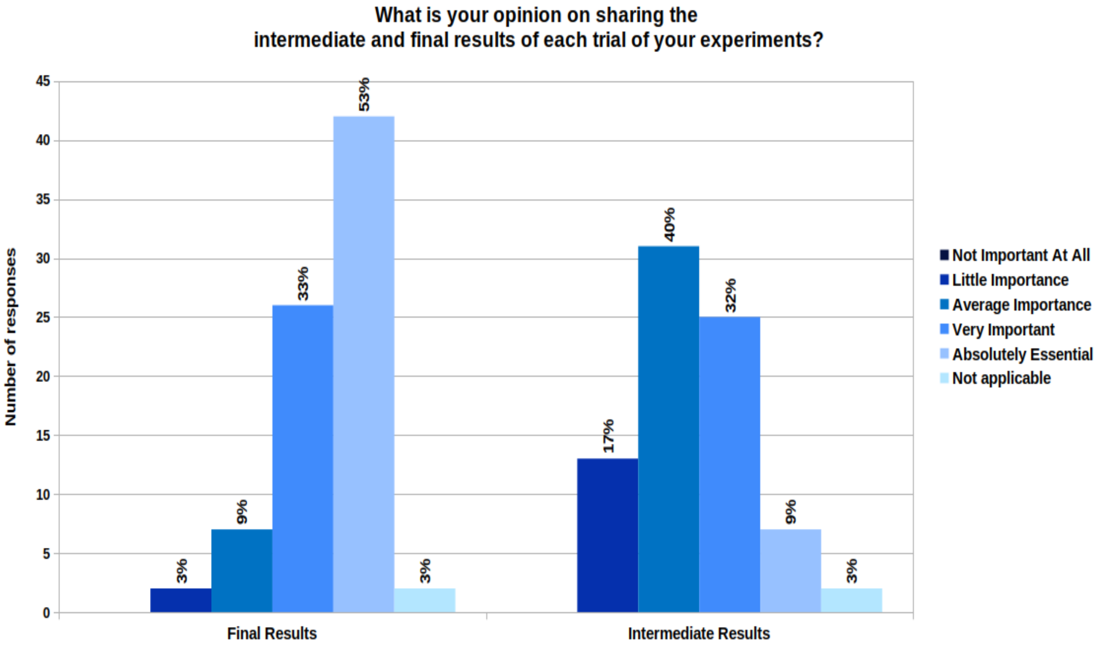


Figure 7.20: What is your opinion on sharing the intermediate and final results of each trial of your experiments?

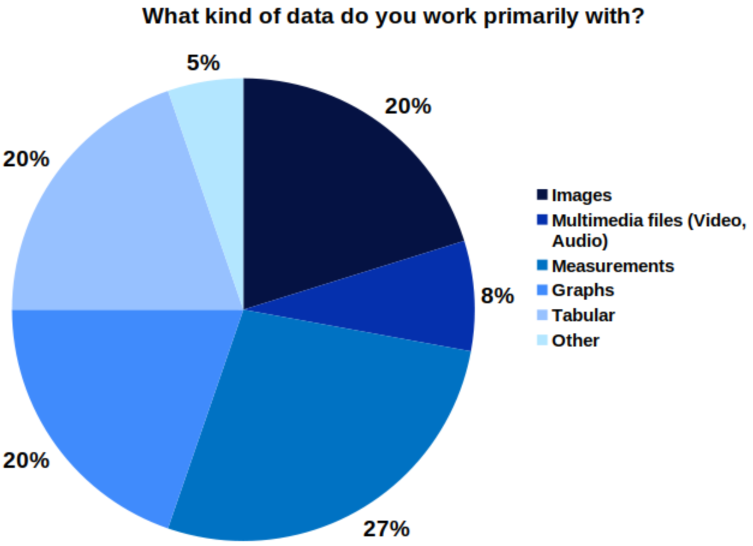


Figure 7.21: What kind of data do you work primarily with?

asked about the storage place for their experimental data files and metadata like descriptions of experiment, methods, samples used, etc. **30%** of them store their experimental data files in the local server provided at their workplace. **25%** store them in their personal devices and **21%** of them specifically store in removable storage devices like hard drive, USB, etc. Only **13%** of them use version-controlled repositories like Github, GitLab, Figshare. Only **8%** of them use data management platforms.

When asked about the experiment metadata storage, **58%** of them use handwritten notebooks as the primary source and **26%** as a secondary source. **51%** of them use electronic notebooks as a primary source and **29%** as secondary source. **54%** of them use data management platforms as either a primary or secondary source.

To know the importance of scripts in the daily research work of researchers, we

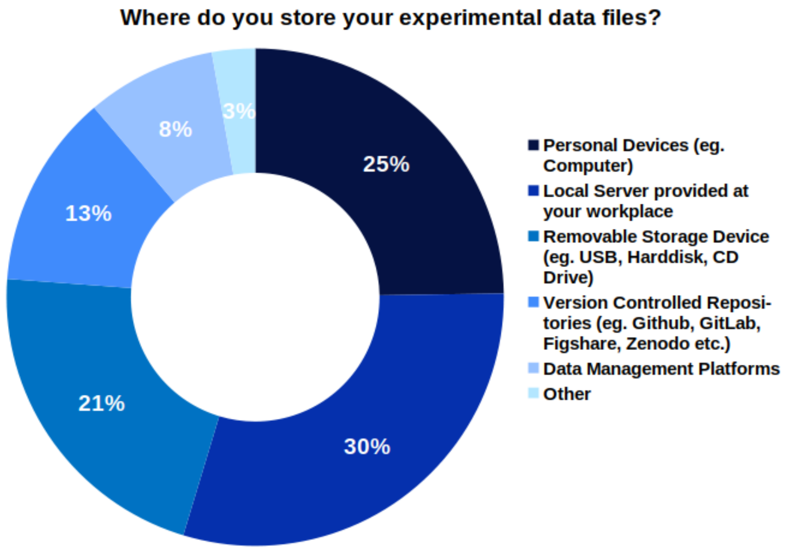


Figure 7.22: Where do you store your experimental data files?

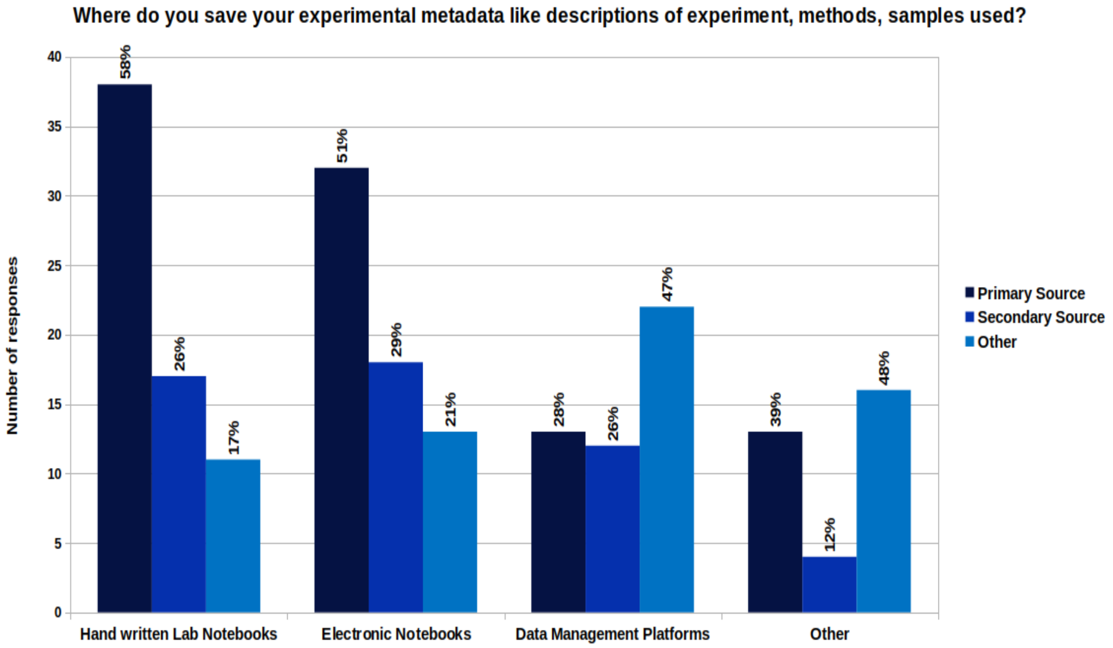


Figure 7.23: Where do you save your experimental metadata like descriptions of experiment, methods, samples used?

asked whether they write programs at any stage in their experimental workflow. **61%** of the participants use scripts or program to perform data analysis. While the other half either use them sometimes (**24%**) or do not use at all (**15%**). So in total, **85%** of participants have used scripts in their experimental workflow. These participants come from not only computer science but also from different other

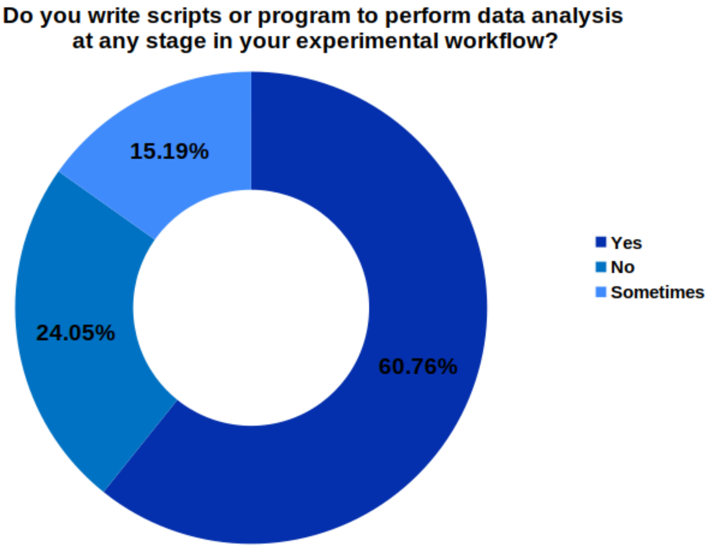


Figure 7.24: Do you write scripts or program to perform data analysis at any stage in your experimental workflow?

scientific fields. In our next questions, we asked the participants about the FAIR principles [Wilkinson et al., 2016]. **62%** of the participants have heard about the FAIR principles and **30%** of them haven't heard about it. **8%** of them have heard the term but do not know exactly what that means. It was interesting to see that

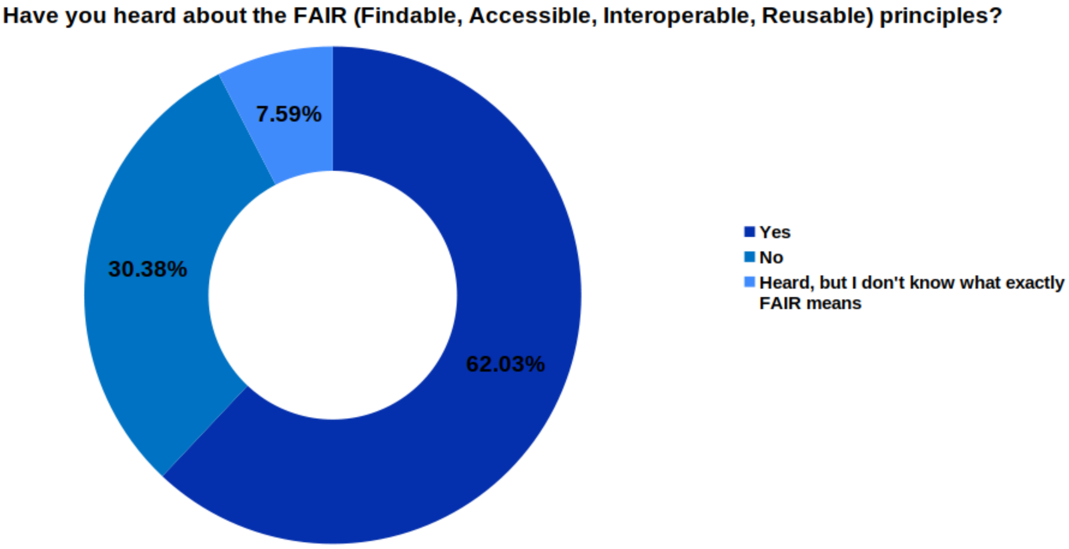


Figure 7.25: Have you heard about the FAIR (Findable, Accessible, Interoperable, Reusable) principles?

the research of the participants are either always or often findable (**72%**), accessible (**69%**), interoperable (**61%**) and reusable (**72%**). We also provided at the end of the survey a free text field to provide comments regarding what they think is important to enable understandability and reproducibility of scientific experiments in their field of research. We got **7** responses which is provided in Appendix C.1.

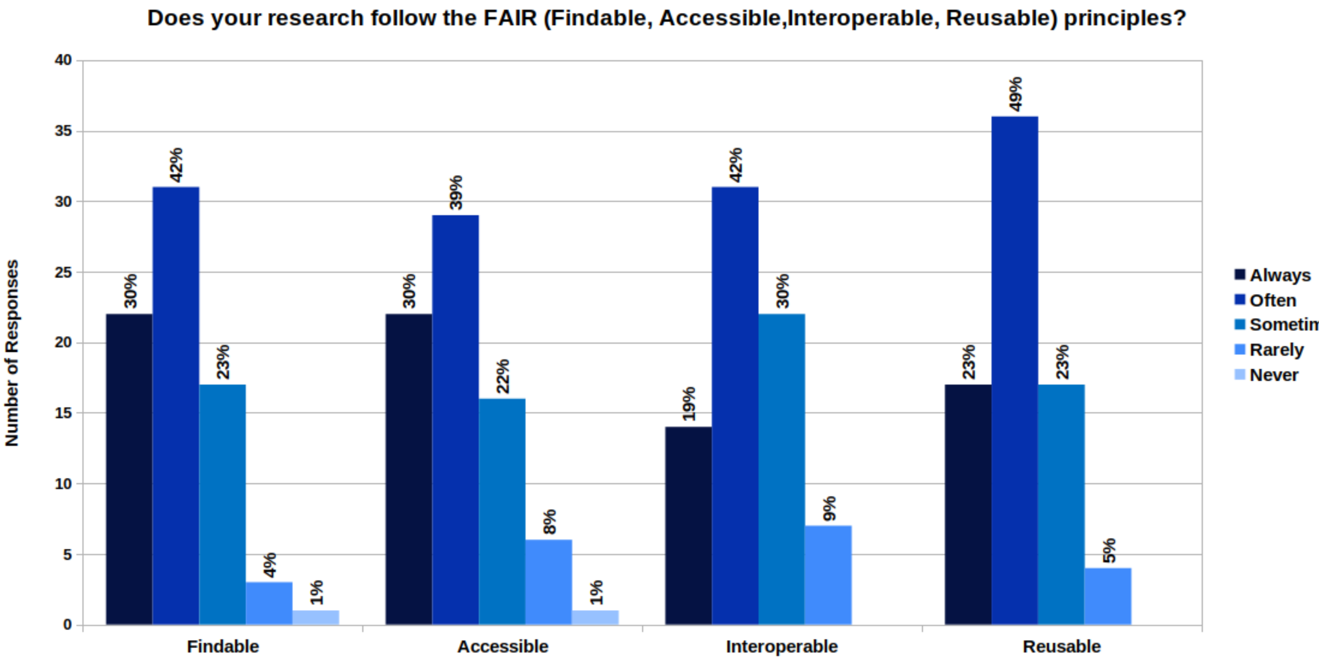


Figure 7.26: Does your research follow the FAIR (Findable, Accessible, Interoperable, Reusable) principles?

7.3.3 Discussion

Analyzing the results from this survey shows that more than half (**59%**) of the participants think that there is a reproducibility crisis. The PhD students and PostDocs who work daily with the data consider it to be an issue. Even though few of the participants said that the 'crisis' is a bad word to say, they agreed that there is a room for improvement and much attention is required to support reproducibility. Lack of sufficient metadata regarding the experiment, lack of data that is publicly available for use and lack of complete information in Method/Standard Operating Procedure/Protocol are considered the important factors that lead to poor reproducibility according to the experience of participants in their research field. Finding metadata about the methods, steps and the experimental setup is considered difficult for both the participants at a later point of time as well as the newcomers in their workplace. Hence, we could see the connection between the lack of complete information in Method/Standard Operating Procedure/Protocol and their findability is considered to be a major factor leading to poor reproducibility. It is also seen from the results that **54%** of the participants had trouble reproducing other's published results. And at the same time, only **5%** of the respondents were contacted about a problem in reproducing their published results. We observe that either people are reluctant to contact the authors or they do not want to take the effort/time to reproduce other's results. It is also seen that **36%** of the participants never tried to reproduce other's published results. Time is considered an issue here since **62%** of the participants think there is lack of time to follow reproducible research practices.

The training on reproducible research practices needs to be provided to scientists. Since the same number of people who think that there is reproducibility crisis also mentioned that there is a lack of training on reproducible research practices (**59%**). Another interesting thing to notice that **53%** of the respondents repeat their own experiments to verify the results while **12%** do not. Therefore, repeatability is important to verify results even if it is at a later point in time. Hence, the data and the steps are important to be documented for both the experimenter as well as the newcomer. With regard to sharing experimental data and metadata, participants are keen to have negative results being shared. Experimental metadata including experiment environment conditions, instruments, and their settings, and experiment materials are also considered important besides results and needs to be shared to ensure reproducibility. We also see that **58%** of the participants use handwritten Laboratory notebooks as their primary source and only **28%** of them use Data management platforms as a primary source. In the current era which is driven by data science, there are more than half of the participants who use the traditional way of documenting experimental metadata. Even though this approach works for people, but it creates difficulty for digital preservation and reproducibility of experiments by the newcomers in the group as pointed earlier. Hence, we require awareness on provenance management of experiments as well. This helps in addressing the issues of the reproducibility crisis. **85%** of the participants use scripts to perform data analysis in their experimental workflow. It indicates the importance of scripts in daily research work of researchers irrespective of their scientific disciplines. Hence, linking script provenance to experimental metadata is required for end-to-end management for reproducibility.

The FAIR principles which came into existence in 2016 is creating an impact on data sharing. We see that **62%** of the participants have heard about the FAIR principles. But **38%** of them haven't heard or do not know exactly what the term means. Another interesting thing to notice is that more than half of the participants have tried to make their research work findable, accessible, interoperable and reusable. However, making research data interoperable by the participants was considered most difficult to follow among the FAIR principles.

Through this survey, we evaluated different aspects of our research work. In order to support end-to-end provenance management, our hypothesis H1 was to design a data model that represents a complete path of scientific experiments. We evaluated the elements that are required for reproducibility in order to represent this complete path in the data model (see Hypothesis H1.1 andH1.2). In Section 7.3.2.4, we evaluated the elements the scientists consider important in sharing experimental data. The elements that we provided in the survey are coming from the REPRODUCE-ME Data Model (see Section 4.4). The important elements of the REPRODUCE-ME Data Model are Experiment, Data, Agent, Activity, Plan, Step, Setting, Instrument, and Material. The results show that each of these elements is considered important or absolutely essential by more than **75%** of the participants. Another thing that is worth mentioning here is that participants considered making their research data interoperable difficult to follow among the FAIR principles. Hence, we can see that the REPRODUCE-ME Ontology addresses the important elements for the end-to-end provenance management of scientific data and also for their interoperability. From the survey results, it is also seen that the findability, accessibility, and reusability of data are difficult not only for their own but also for the newcomers in the team. This gets more difficult for the metadata about the methods, steps, and experiment setup. Hence, these results clearly point out that the FAIR principles need to be followed from the bottom level of the research lifecycle (see Figure 1.1). The scientific data management platforms like CAESAR help to address this issue by providing a provenance-based semantic and collaborative approach for the management of experimental data.

7.4 Computational Reproducibility

In this section, we evaluate how ProvBook supports computational reproducibility using Jupyter Notebooks. We use data and user-based evaluation to validate our hypothesis H3-H4. We did user-based evaluation in conjunction with a master student (Bastian Bunzeck) as part of Semantic Web Technologies Course[13] at University of Jena, Germany. This evaluation was done with Jupyter notebooks which are publicly available. It focused on how ProvBook performs with different aspects of usability, performance, and scalability in addition to reproducibility. We did a study to see how the provenance capture, visualization and difference provided by ProvBook help in different use case scenarios to support computational reproducibility. Random Jupyter notebooks were collected from github and evaluated with ProvBook. Here we show the evaluation with one such Jupyter Notebook. The evaluation was done based on the following factors and scenarios:

1. A notebook executed by two different users.
2. A notebook executed by two different users in different environments.
3. The input, output, execution time and the order in two different executions of a notebook.
4. Provenance difference of the results of a notebook.
5. Performance of ProvBook with respect to time.
6. Performance of ProvBook with respect to space.

[13] `https://caj.informatik.uni-jena.de/caj/course/details/id/-310264951709084758`

Notebook Name	User	Environment
eigenfaces	Original Author	Scikit-learn 0.16
eigenfaces	User 1	Ubuntu 18.10, Scikit-learn 0.20.0, Python 3
eigenfaces	User 1	Fedora, Scikit-learn 0.20.0, Python 3
eigenfaces	User 2	Ubuntu 18.04, Scikit-learn 0.20.3, Python 2
eigenfaces	User 2	Ubuntu 18.04, Scikit-learn 0.20.3, Python 3

Table 7.1: The statistics of the Jupyter Notebook executions

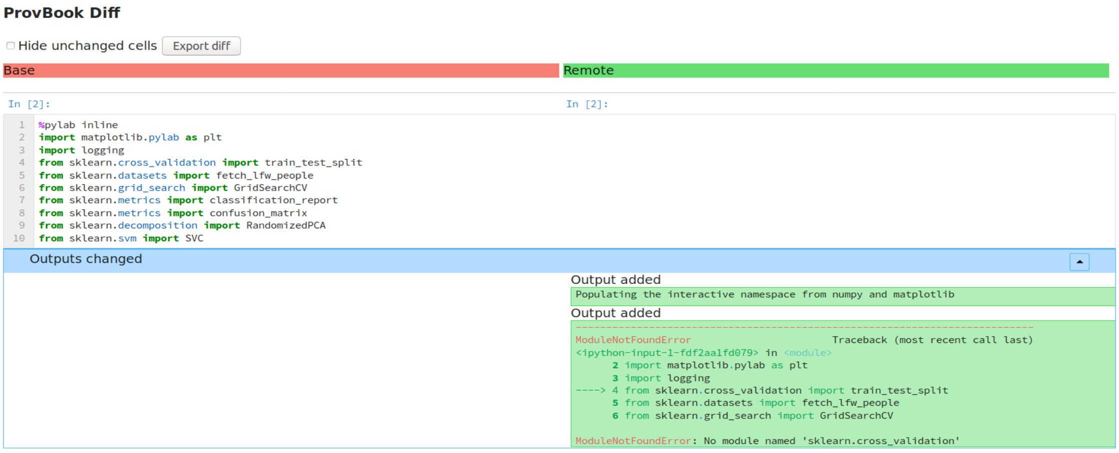
7. Complete path taken by a computational experiment with the sequence of steps in the execution of a notebook with input parameters and intermediate results in each step required to generate the final output.

8. The environmental attributes in the execution of a notebook.

We use an example Jupyter Notebook which uses face recognition example applying eigenface algorithm and SVM using scikit-learn [Pedregosa et al., 2011]. The initial code is adapted from scikit-learn[14]. We use *Original Author* to refer to the author who is the first author of the notebook and *User 1* and *User 2* to the authors who used the original notebook to reproduce results. The notebook was first saved without any outputs. Later the notebook was executed by two different users. The notebook was run in three different environments. Table 7.1 provides the statistics of the Jupyter Notebook executions. The first run of the eigenfaces Jupyter Notebook gave ModuleNotFoundError for *User 1*. Several runs were attempted to solve the issue. However, for *User 2*, only the first run gave ModuleNotFoundError error. This was resolved by installing the scikit-learn module. But for *User 1* installing the module still did not solve the issue. The problem occurred because of the version change of the scikit-learn module. The original Jupyter Notebook used 0.16 version of scikit-learn. While *User 1* used 0.20.0 version, *User 2* used 0.20.3. The classes and functions from the cross_validation, grid_search, and learning_curve modules were placed into a new model_selection module starting from Scikit-learn 0.18. Several other changes were made in the script which used these functions. *User 1* made the necessary changes to work for the new versions of the scikit-learn module, hence, *User 2* did not have to change scripts. Using ProvBook, Users 1 and 2 could track the changes and compare the original script with the new one which worked on both the user's systems. Figures 7.27 and 7.28 show the differences of the several runs to reproduce the results of eigenfaces Jupyter notebook.

Figures 7.29, 7.30 and 7.31 show the different execution times for the fourth cell in the notebook in different environments. The fourth cell consists of a function $fetch\_lfw\_people$ which downloads a set of preprocessed images if they are not al-

[14] https://scikit-learn.org/0.16/_downloads/face_recognition.py



In [2]:

```
%pylab inline
import matplotlib.pylab as plt
import logging
from sklearn.cross_validation import train_test_split
from sklearn.datasets import fetch_lfw_people
from sklearn.grid_search import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.decomposition import RandomizedPCA
from sklearn.svm import SVC
```

In [2]:

Outputs changed

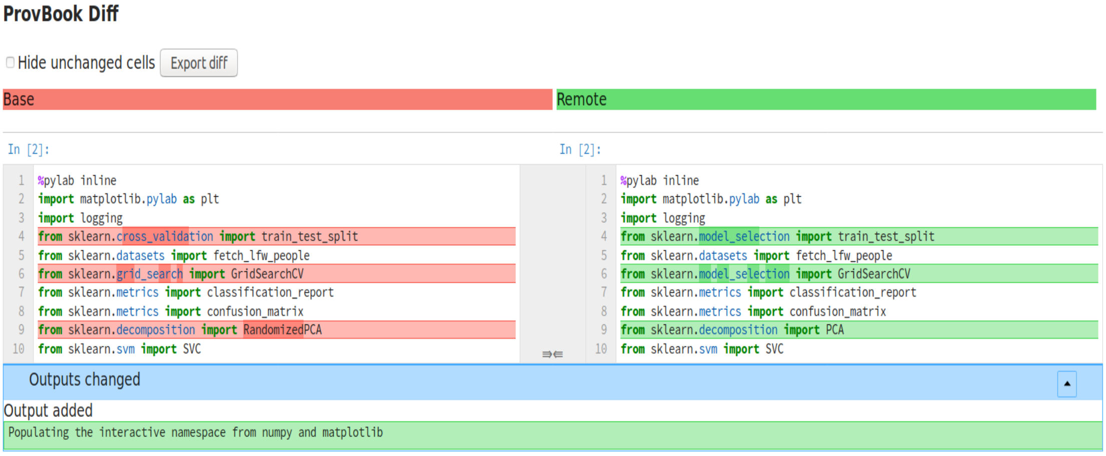
Output added

```
Populating the interactive namespace from numpy and matplotlib
```

Output added

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-1-fdf2aa1fd079> in <module>
      2 import matplotlib.pylab as plt
      3 import logging
----> 4 from sklearn.cross_validation import train_test_split
      5 from sklearn.datasets import fetch_lfw_people
      6 from sklearn.grid_search import GridSearchCV

ModuleNotFoundError: No module named 'sklearn.cross_validation'
```

Figure 7.27: Comparison of the first cell from the original author with the first execution by User 1



In [2]:

```
%pylab inline
import matplotlib.pylab as plt
import logging
from sklearn.cross_validation import train_test_split
from sklearn.datasets import fetch_lfw_people
from sklearn.grid_search import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.decomposition import RandomizedPCA
from sklearn.svm import SVC
```

In [2]:

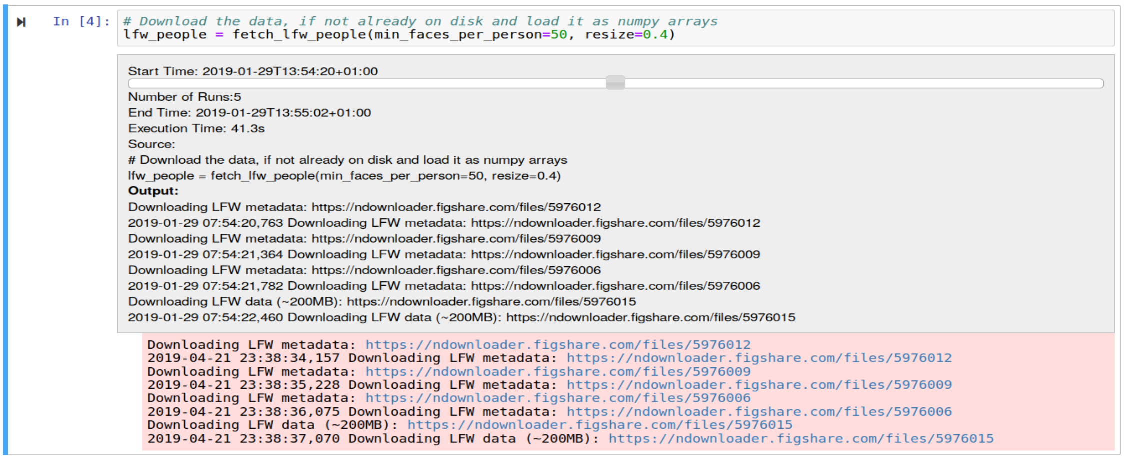
```
%pylab inline
import matplotlib.pylab as plt
import logging
from sklearn.model_selection import train_test_split
from sklearn.datasets import fetch_lfw_people
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.decomposition import PCA
from sklearn.svm import SVC
```

Outputs changed

Output added

```
Populating the interactive namespace from numpy and matplotlib
```

Figure 7.28: Comparison of the first cell from the original author after making changes in the fifth execution by User 1

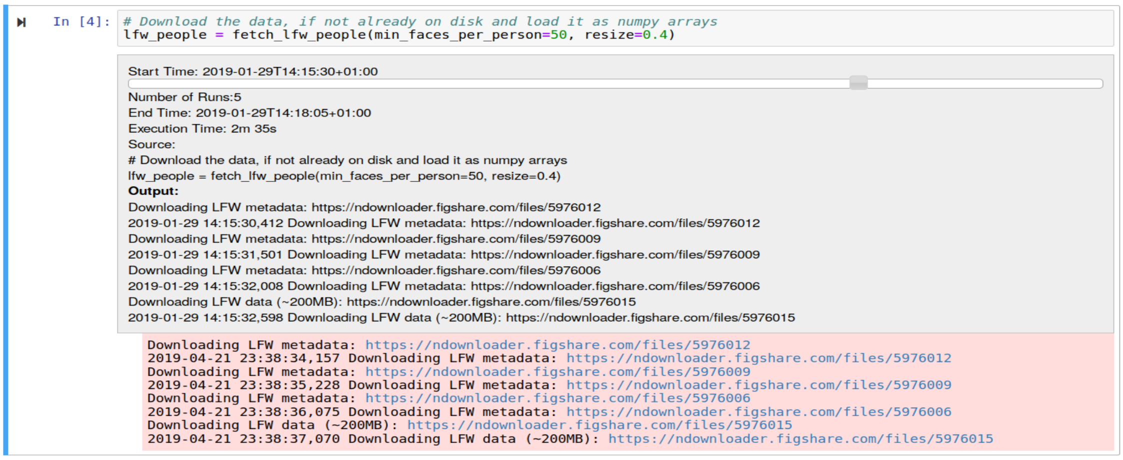


```
In [4]: # Download the data, if not already on disk and load it as numpy arrays
        lfw_people = fetch_lfw_people(min_faces_per_person=50, resize=0.4)
```

```
Start Time: 2019-01-29T13:54:20+01:00
Number of Runs:5
End Time: 2019-01-29T13:55:02+01:00
Execution Time: 41.3s
Source:
# Download the data, if not already on disk and load it as numpy arrays
lfw_people = fetch_lfw_people(min_faces_per_person=50, resize=0.4)
Output:
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976012
2019-01-29 07:54:20,763 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976012
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976009
2019-01-29 07:54:21,364 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976009
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976006
2019-01-29 07:54:21,782 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976006
Downloading LFW data (~200MB): https://ndownloader.figshare.com/files/5976015
2019-01-29 07:54:22,460 Downloading LFW data (~200MB): https://ndownloader.figshare.com/files/5976015
```

```
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976012
2019-04-21 23:38:34,157 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976012
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976009
2019-04-21 23:38:35,228 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976009
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976006
2019-04-21 23:38:36,075 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976006
Downloading LFW data (~200MB): https://ndownloader.figshare.com/files/5976015
2019-04-21 23:38:37,070 Downloading LFW data (~200MB): https://ndownloader.figshare.com/files/5976015
```

Figure 7.29: Execution time for the fourth cell in the third run by User 1 in first environment



```
In [4]: # Download the data, if not already on disk and load it as numpy arrays
        lfw_people = fetch_lfw_people(min_faces_per_person=50, resize=0.4)
```

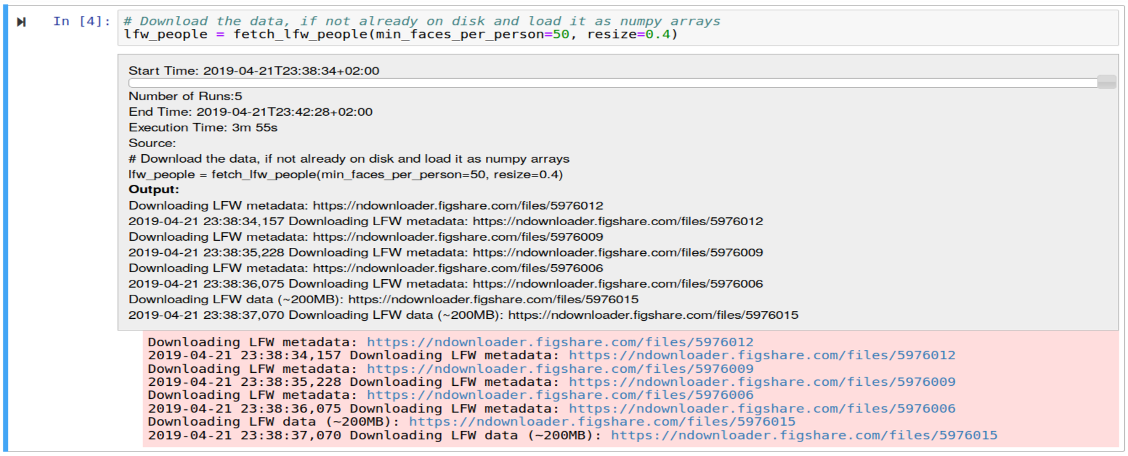
```
Start Time: 2019-01-29T14:15:30+01:00
Number of Runs:5
End Time: 2019-01-29T14:18:05+01:00
Execution Time: 2m 35s
Source:
# Download the data, if not already on disk and load it as numpy arrays
lfw_people = fetch_lfw_people(min_faces_per_person=50, resize=0.4)
Output:
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976012
2019-01-29 14:15:30,412 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976012
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976009
2019-01-29 14:15:31,501 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976009
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976006
2019-01-29 14:15:32,008 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976006
Downloading LFW data (~200MB): https://ndownloader.figshare.com/files/5976015
2019-01-29 14:15:32,598 Downloading LFW data (~200MB): https://ndownloader.figshare.com/files/5976015
```

```
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976012
2019-04-21 23:38:34,157 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976012
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976009
2019-04-21 23:38:35,228 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976009
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976006
2019-04-21 23:38:36,075 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976006
Downloading LFW data (~200MB): https://ndownloader.figshare.com/files/5976015
2019-04-21 23:38:37,070 Downloading LFW data (~200MB): https://ndownloader.figshare.com/files/5976015
```

Figure 7.30: Execution time for the fourth cell in the fourth run by User 1 in second environment

ready present in the disk. It downloads data from Labeled Faces in the World (LFW)[15] which contains the training data for face recognition study. We could see that in Figure 7.29, it took around 41.3ms in the first environment for the complete execution of the cell while in Figure 7.31, it increased to 3min 55s in the third environment. The different execution environments clearly play a role in computational experiments which is clearly shown with the help of ProvBook. Figure 7.32 shows the difference in the intermediate result (the quantative evaluation of the model quality on the test data) in two different executions by two different users in two different environments. There is no change in the input of the cell in both executions, however, the change in the previous cell affected the results. Figure 7.33 shows the difference in the input in the two different executions of the cells which caused the change in the results. The provenance capture and difference in ProvBook can handle different types of output including images. Figure 7.34 shows such case displaying the difference of a cell execution in the images. This evaluation was done with several output types mentioned in Section 5.1.

[15] http://vis-www.cs.umass.edu/lfw/



In [4]:

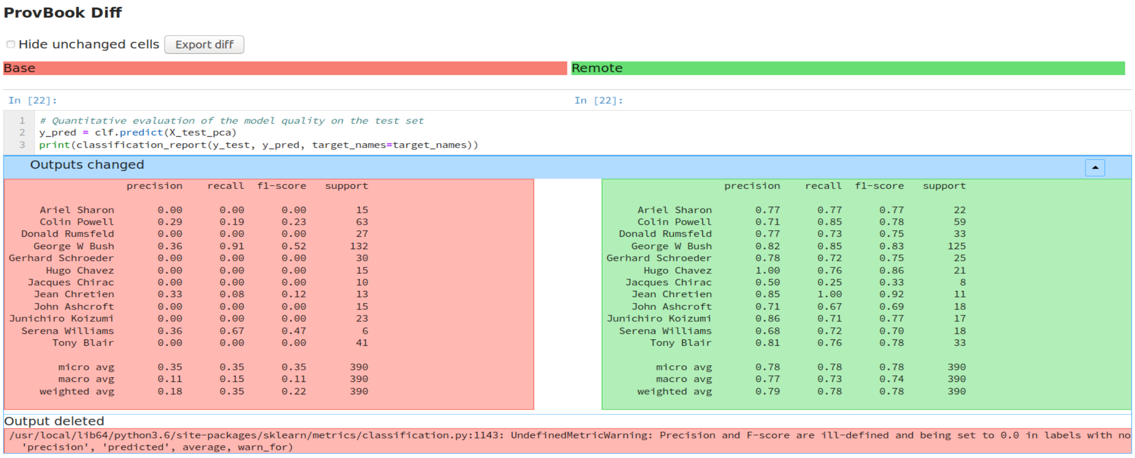
```
# Download the data, if not already on disk and load it as numpy arrays
lfw_people = fetch_lfw_people(min_faces_per_person=50, resize=0.4)
```

Start Time: 2019-04-21T23:38:34+02:00

Number of Runs:5
End Time: 2019-04-21T23:42:28+02:00
Execution Time: 3m 55s
Source:
# Download the data, if not already on disk and load it as numpy arrays
lfw_people = fetch_lfw_people(min_faces_per_person=50, resize=0.4)
Output:
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976012
2019-04-21 23:38:34,157 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976012
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976009
2019-04-21 23:38:35,228 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976009
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976006
2019-04-21 23:38:36,075 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976006
Downloading LFW data (~200MB): https://ndownloader.figshare.com/files/5976015
2019-04-21 23:38:37,070 Downloading LFW data (~200MB): https://ndownloader.figshare.com/files/5976015

```
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976012
2019-04-21 23:38:34,157 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976012
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976009
2019-04-21 23:38:35,228 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976009
Downloading LFW metadata: https://ndownloader.figshare.com/files/5976006
2019-04-21 23:38:36,075 Downloading LFW metadata: https://ndownloader.figshare.com/files/5976006
Downloading LFW data (~200MB): https://ndownloader.figshare.com/files/5976015
2019-04-21 23:38:37,070 Downloading LFW data (~200MB): https://ndownloader.figshare.com/files/5976015
```

Figure 7.31: Execution time for the fourth cell in the fifth run by User 2 in third environment



ProvBook Diff

□ Hide unchanged cells

In [22]:

```
# Quantitative evaluation of the model quality on the test set
y_pred = clf.predict(X_test_pca)
print(classification_report(y_test, y_pred, target_names=target_names))
```

Outputs changed

```
                   precision    recall  f1-score   support

     Ariel Sharon       0.00      0.00      0.00        15
     Colin Powell       0.29      0.19      0.23        63
  Donald Rumsfeld       0.00      0.00      0.00        27
    George W Bush       0.36      0.91      0.52       132
Gerhard Schroeder       0.00      0.00      0.00        30
      Hugo Chavez       0.00      0.00      0.00        15
   Jacques Chirac       0.00      0.00      0.00        10
    Jean Chretien       0.33      0.08      0.12        13
    John Ashcroft       0.00      0.00      0.00        15
Junichiro Koizumi       0.00      0.00      0.00        23
  Serena Williams       0.36      0.67      0.47         6
       Tony Blair       0.00      0.00      0.00        41

        micro avg       0.35      0.35      0.35       390
        macro avg       0.11      0.15      0.11       390
     weighted avg       0.18      0.35      0.22       390
```

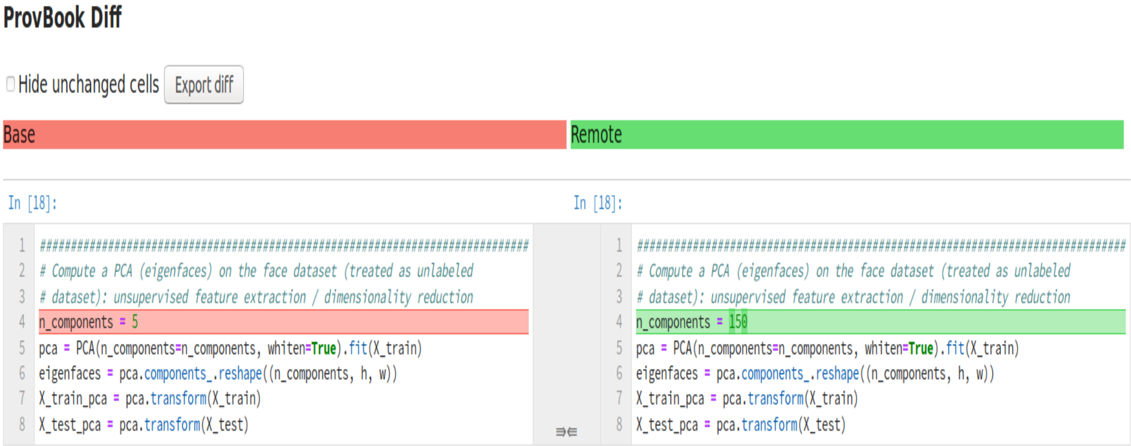
In [22]:

```
                   precision    recall  f1-score   support

     Ariel Sharon       0.77      0.77      0.77        22
     Colin Powell       0.71      0.85      0.78        59
  Donald Rumsfeld       0.77      0.73      0.75        33
    George W Bush       0.82      0.85      0.83       125
Gerhard Schroeder       0.78      0.72      0.75        25
      Hugo Chavez       1.00      0.76      0.86        21
   Jacques Chirac       0.50      0.25      0.33         8
    Jean Chretien       0.85      1.00      0.92        11
    John Ashcroft       0.71      0.67      0.69        18
Junichiro Koizumi       0.86      0.71      0.77        17
  Serena Williams       0.68      0.72      0.70        18
       Tony Blair       0.81      0.76      0.78        33

        micro avg       0.78      0.78      0.78       390
        macro avg       0.77      0.73      0.74       390
     weighted avg       0.79      0.78      0.78       390
```

Output deleted

```
/usr/local/lib64/python3.6/site-packages/sklearn/metrics/classification.py:1143: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no
  'precision', 'predicted', average, warn_for)
```

Figure 7.32: The difference in the output without any change in the input of the cell



ProvBook Diff

□ Hide unchanged cells

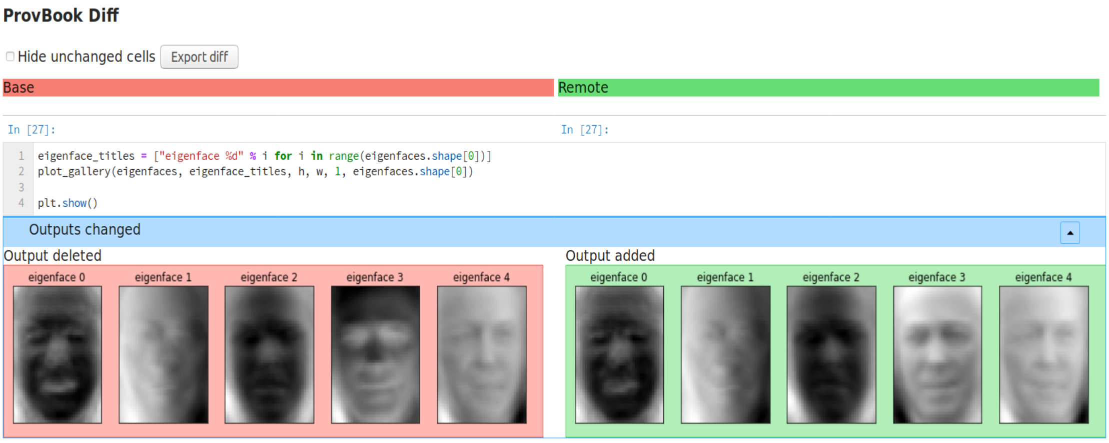
In [18]:

```
###############################################################################
# Compute a PCA (eigenfaces) on the face dataset (treated as unlabeled
# dataset): unsupervised feature extraction / dimensionality reduction
n_components = 5
pca = PCA(n_components=n_components, whiten=True).fit(X_train)
eigenfaces = pca.components_.reshape((n_components, h, w))
X_train_pca = pca.transform(X_train)
X_test_pca = pca.transform(X_test)
```

In [18]:

```
###############################################################################
# Compute a PCA (eigenfaces) on the face dataset (treated as unlabeled
# dataset): unsupervised feature extraction / dimensionality reduction
n_components = 150
pca = PCA(n_components=n_components, whiten=True).fit(X_train)
eigenfaces = pca.components_.reshape((n_components, h, w))
X_train_pca = pca.transform(X_train)
X_test_pca = pca.transform(X_test)
```

Figure 7.33: The difference in the input with modification in the input of the cell



ProvBook Diff

□ Hide unchanged cells Export diff

Base | Remote

In [27]:

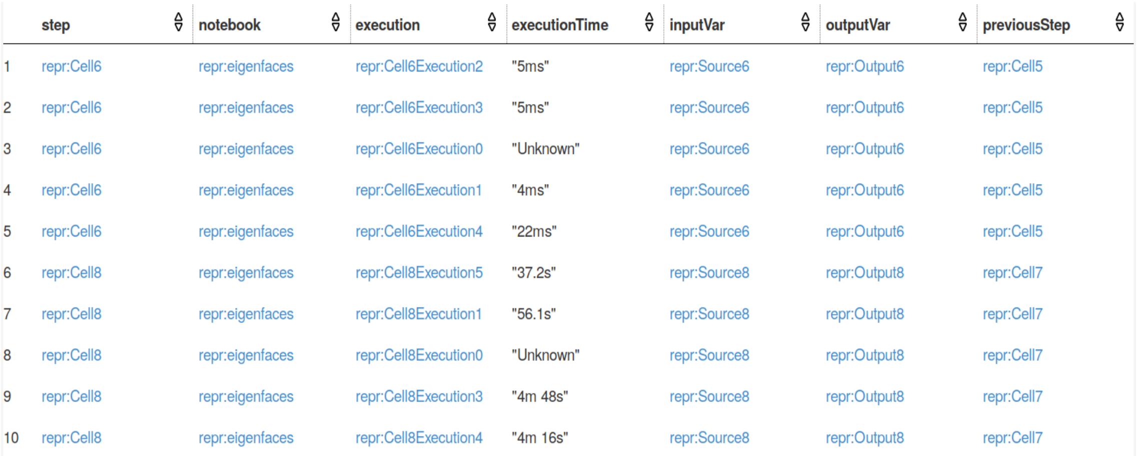
```
eigenface_titles = ["eigenface %d" % i for i in range(eigenfaces.shape[0])]
plot_gallery(eigenfaces, eigenface_titles, h, w, 1, eigenfaces.shape[0])

plt.show()
```

Outputs changed

Output deleted

Output added

Figure 7.34: The difference in the results which are images



	step	notebook	execution	executionTime	inputVar	outputVar	previousStep
1	repr:Cell6	repr:eigenfaces	repr:Cell6Execution2	"5ms"	repr:Source6	repr:Output6	repr:Cell5
2	repr:Cell6	repr:eigenfaces	repr:Cell6Execution3	"5ms"	repr:Source6	repr:Output6	repr:Cell5
3	repr:Cell6	repr:eigenfaces	repr:Cell6Execution0	"Unknown"	repr:Source6	repr:Output6	repr:Cell5
4	repr:Cell6	repr:eigenfaces	repr:Cell6Execution1	"4ms"	repr:Source6	repr:Output6	repr:Cell5
5	repr:Cell6	repr:eigenfaces	repr:Cell6Execution4	"22ms"	repr:Source6	repr:Output6	repr:Cell5
6	repr:Cell8	repr:eigenfaces	repr:Cell8Execution5	"37.2s"	repr:Source8	repr:Output8	repr:Cell7
7	repr:Cell8	repr:eigenfaces	repr:Cell8Execution1	"56.1s"	repr:Source8	repr:Output8	repr:Cell7
8	repr:Cell8	repr:eigenfaces	repr:Cell8Execution0	"Unknown"	repr:Source8	repr:Output8	repr:Cell7
9	repr:Cell8	repr:eigenfaces	repr:Cell8Execution3	"4m 48s"	repr:Source8	repr:Output8	repr:Cell7
10	repr:Cell8	repr:eigenfaces	repr:Cell8Execution4	"4m 16s"	repr:Source8	repr:Output8	repr:Cell7

Figure 7.35: Complete path taken by a user for a computational notebook experiment

We also evaluated the performance of ProvBook with respect to space and time. Regarding time, the difference in the execution time of each cell with and without ProvBook was negligible. Regarding space, the size of the Jupyter Notebook with provenance information of several executions was more than the original notebook. As stated in [Chapman et al., 2008], the size of the provenance information can grow more than the actual data.

In the next scenario, we evaluated the semantic representation of the provenance of computational notebooks and scripts. Listings 7.1 shows the SPARQL query of the complete path taken by a computational experiment with input parameters and intermediate results in each step required to generate the final output. Figure 7.35 shows the result from this query for the competency question **CQ11**. It also shows the sequence of steps in the execution of the notebook. SPARQL Query 7.2 is responsible for querying the environmental attributes of notebooks in different execution environments. The results of this query is shown in Table 7.2 for the competency question **CQ19**.

```
SELECT DISTINCT * WHERE
{
```

notebook	ProgrammingLanguage	version	Kernel
eigenfaces	python	2.7.15rc1	python3
eigenfaces	python	3.6.8	python3

Table 7.2: The environmental attributes of a notebook's execution

```
    ?step p-plan:isStepOfPlan ?notebook .
    ?notebook a repr:Notebook .
    ?execution p-plan:correspondsToStep ?step ;
      repr:executionTime ?executionTime .
    ?step p-plan:hasInputVar ?inputVar ;
      p-plan:hasOutputVar ?outputVar ;
      p-plan:isPrecededBy ?previousStep .
}
```

Listing 7.1: Complete path for a computational notebook experiment

```
SELECT DISTINCT * WHERE
{
    ?notebook a :Notebook ;
    :hasProgrammingLanguage ?ProgrammingLanguage ;
    :hasProgrammingLanguageVersion ?version ;
    :hasKernelName ?Kernel .
}
```

Listing 7.2: Execution environment attributes of computational experiment

To evaluate the semantic representation of scripts using the REPRODUCE-ME ontology, we collected the provenance data of the execution of scripts using the noWorkflow [Murta et al., 2014]. The noWorkflow tool captures provenance of a script by running the command "now run <script>". The provenance data is stored in SQLite relational database in the same directory where the script is executed. The noWorkflow captures information of each run of a script, the function definitions, start and finish time of each trial and activation of the function. The provenance data captured from the execution of a script using noWorkflow tool are populated in the database tables and mapped to the ontology. Listing 7.3 represents the mapping for a trial, FunctionActivation and the sequence of `p-plan:Step` in a trial of a script.

```
mappingId Trial
target  :trial/{id} a :Trial ; prov:value {id} ; prov:startedAtTime
    {start} ; prov:endedAtTime {finish} .
source  select id, start, finish from trial

mappingId Function Activation
target  :activation/{trial_id}/{id} a :FunctionActivation; :name {
   name}; prov:startedAtTime {start} ; prov:endedAtTime {finish} .
source  select trial_id, id, name, start, finish from
   function_activation

```

```
mappingId Step preceded by another Step
target  :activation/{trial_id}/{id} p-plan:isPrecededBy
:activation/{trial_id}/{caller_id} .
source  select trial_id, id, caller_id from function_activation
```

Listing 7.3: Mappings for script execution

We evaluate our approach by using REPRODUCE-ME ontology to answer the competency questions defined in Section 5.3. Here we present two example SPARQL queries related to the provenance information of script execution.
The SPARQL Query for competency question **CQ24** is listed in Listing 7.4.

```
SELECT DISTINCT ?function_2_name ?function_1_name ?output_val WHERE
    {
  ?function_1 a :FunctionActivation ; :name ?function_1_name ;
   prov:startedAtTime ?started_at ;  :correspondsToActivity ?trial
   .
  ?function_2 a :FunctionActivation ; :name ?function_2_name .
  ?function_1 p-plan:isPrecededBy ?function_2 .
  ?output p-plan:isOutputVarOf ?function_1 ; prov:value ?output_val
    .
  ?trial a :Trial ; prov:used ?script ; prov:value ?trial_id FILTER
   (?trial_id="2"^^xsd:integer) .
  ?script :name ?script_name .
  }
ORDER by ?started_at
```

Listing 7.4: The complete derivation of a script output

The SPARQL Query for competency question **CQ23** is listed in Listing 7.5.

```
SELECT DISTINCT * WHERE {
  ?os a :OperatingSystem ; :name ?os_name ; :version ?os_version .
  ?trial a :Trial ; prov:used ?os ;
    prov:atLocation ?execution_directory ;
    prov:wasStartedBy ?experimenter .
  ?experimenter a :Experimenter ; :name ?experimenter_name .
  ?trial prov:used ?pl .
  ?pl a :ProgrammingLanguage ;
    :name ?programming_language ;
    :version ?programming_language_version .
}
```

Listing 7.5: List the environment attributes of the execution of a script

Table 7.3 shows the results from the execution of a script “factorial.py” which calculates the factorial of a number for the SPARQL Query 7.4. The script “factorial.py” is executed twice with the same input 5 and same environment attributes like operating system, programming language version, processor etc. We see that the two trials of a script under the same execution environment and same input parameters in each step follows the same path to generate the same final output which is 120. The environmental settings are also required for the reproducibility of experimental

function_2_name	function_1_name	output_val
factorial.py	main	None
main	factorial	120
factorial	factorial	24
factorial	factorial	6
factorial	factorial	2
factorial	factorial	1
main	print_message	None

Table 7.3: Result for SPARQL Query 7.4

data. Here we do not take the randomness factor of input parameters into consideration. Also, we do not consider line by line execution of script rather focus on functions as steps.

7.4.1 Discussion

This section focused on the evaluation of our work in supporting computational reproducibility. Here, we targeted only computational experiments using computational notebooks and scripts. The results of the data and user-based evaluation clearly shows how ProvBook helps in supporting computational reproducibility. We see that how each item added in the provenance information in Jupyter Notebooks helps to track the changes in the results even in different execution environments. The input, output, starting and ending time, and the execution time for each trial from each experimenter helps in tracking the provenance of the computational experiments. The Jupyter Notebooks shared along with the provenance information of their executions helps to compare the original intermediate and final results with the results from the new trials executed in the same or different environment. We see that it not only helps in reproducibility (Definition 4.1.4) but also with repeatability (Definition 4.1.5). This helps in tracking the intermediate and negative results and the input and the output from different trials are not lost. The execution environmental attributes of the computational experiments along with their results help to understand their complete path. This solves two of the problems faced by Ana described in our use-case in Section 2.1. Hence, it validates the hypothesis H3 and H4. The results also show that ProvBook can store provenance of different types of output of each cell. We also see that we could describe the relationship between the results, the execution environment and the executions that generated the results of a computational experiment in an interoperable way using the REPRODUCE-ME ontology. Hence, it validates the hypothesis H1.2 and H2.

7.5 CAESAR

We evaluate our research work with real-life scientific experiments. Application-based evaluation is one of the approaches to evaluating ontologies [Brank et al., 2005] which will be used in our case. In application-based evaluation, the ontology under evaluation is used in an application/system to produce good results on a given task. We plug our ontology into CAESAR to describe the provenance of scientific experiments. Users are also involved in the evaluation as well as being the consumers of our system. The evaluation was done on a server hosted at the University Hospital Jena. The server is installed with CentOS Linux 7 and has x86-64 architecture (Intel Corporation Xeon E7 v3/Xeon E5 v3/Core i7). The storage of the system is divided into two components: 150 TB archive with large access time and 100TB for faster access time. It has 16 GB RAM. CAESAR is installed in the system with OMERO, JupyterHub, and ProvBook.

We evaluated the REPRODUCE-ME ontology and CAESAR in the context of scientific experiments related to high-end light microscopy. Scientists from B1 and A4 projects of ReceptorLight used and evaluated the system. Experiments using confocal patch-clamp fluorometry (cPCF), Förster Resonance Energy Transfer (FRET), PhotoActivated Localization Microscopy (PALM) and direct Stochastic Optical Reconstruction Microscopy (dSTORM) were documented by the scientists as part of their daily work. Total of 44 experiments in 23 projects were recorded (Accessed on April 21, 2019). 373 microscopy images generated from different instruments with various settings were uploaded to the system. The images amount to 15.4 GB of storage. The datasets were uploaded and documented by the scientists using either the desktop client or webclient of CAESAR. The scientific experiments along with the steps, experiment materials, settings, and standard operating procedures were described using the REPRODUCE-ME ontology using Ontology-based data access (OBDA). Table 7.4 shows the statistics of the datasets in CAESAR used for evaluation. In addition to these, we also used another dataset for evaluation which is from the Image Data Repository (IDR) with around 35 imaging experiments[16] [Williams et al., 2017]. This was done to ensure that the REPRODUCE-ME ontology can be used to describe other types of experiments as well. The metadata from each imaging experiment from IDR was extracted and described in RDF using the REPRODUCE-ME ontology. We created a knowledge base of different types of experiments from these two sources.

We first evaluate the REPRODUCE-ME ontology using competency questions. Later, we show how we plugged the ontology in CAESAR by using these competency questions for the visualization of the complete path of a scientific experiment.

We use the concept of *Competency Questions* to validate our hypothesis. Compe-

[16] `https://github.com/IDR/idr-metadata`, Accessed on August 21, 2018

Item	Count
Experiment	44
Project	23
Dataset	60
Image	373
Plasmid	7
Protein	5
Vector	3
Chemical	10
Standard Operating Procedure	5
RNA	1
DNA	1
Fluorescent Protein	1
Oligonucleotide	6
Restriction Enzyme	3
Solution	10

Table 7.4: The statistics of the datasets uploaded and documented in CAESAR

tency questions are the basis for the development of an ontology in determining its scope. The ontology should be able to answer the competency questions over a knowledge base [Noy et al., 2001]. Based on this, we provided a list of competency questions in Section 4.2. To see if the ontology can answer the questions, we generated SPARQL queries for each of the competency questions and executed them on our knowledge base which consists of linked data in CAESAR. The competency questions were translated into SPARQL queries by computer scientists. The domain experts evaluated the correctness of the answers for these competency questions. The competency questions, the RDF data used for the evaluation, the SPARQL queries, and their results are publicly available [Samuel, 2019].

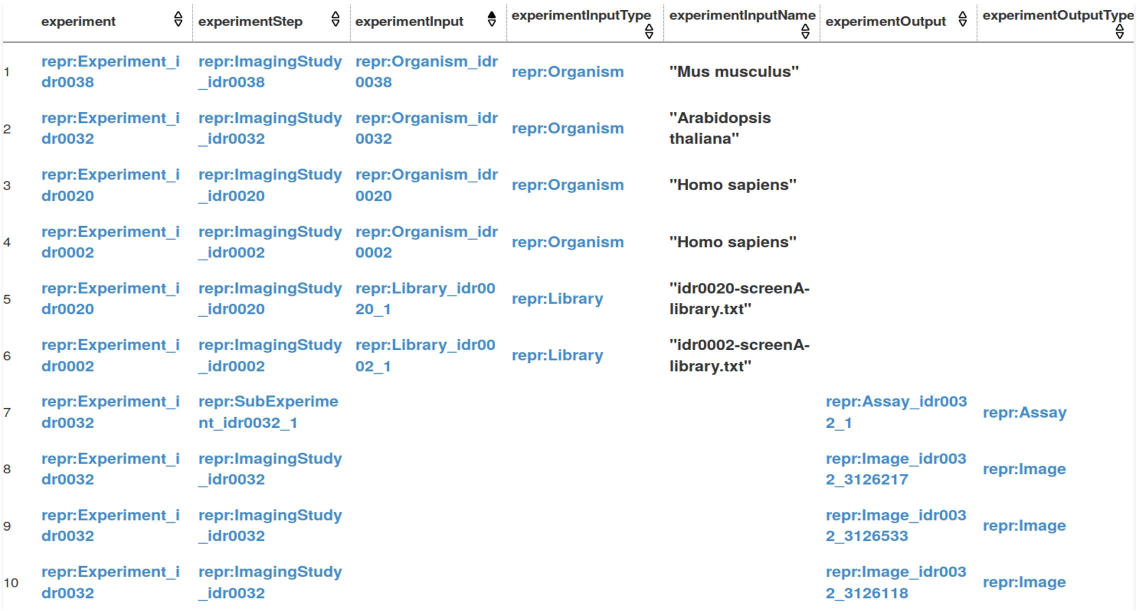
We present the competency questions with the corresponding SPARQL queries and part of the results obtained on running them against the knowledge base. The result of each query is a long list of values, hence, we show only the first few rows from them.

SQ1: What are the input and output variables of an experiment?

This query is responsible for getting both the input and output data of an experiment. Here an experiment is considered as a plan which consists of several steps. The experiment can also have sub plans (e.g. Jupyter Notebook). Each step has input and output variables. If we want to get the values for a particular experiment or a particular step, we can use the **FILTER** keyword in the SPARQL query. Figure 7.36 presents the results of this query showing the experiment, the steps, the input and output of each step.

```
SELECT DISTINCT * WHERE
{
    ?experiment a repr:Experiment .
    ?experimentStep p-plan:isStepOfPlan ?experiment .
    {
      ?experimentInput p-plan:isInputVarOf ?experimentStep ;
                        rdf:type ?experimentInputType .
      OPTIONAL {
        ?experimentInput repr:name ?experimentInputName
      }
    }
    UNION {
      ?experimentOutput p-plan:isOutputVarOf ?experimentStep ;
                        rdf:type ?experimentOutputType .
      OPTIONAL {
        ?experimentOutput repr:name ?experimentOutputName }
      }
}
```

Listing 7.6: SPARQL Query 1



	experiment	experimentStep	experimentInput	experimentInputType	experimentInputName	experimentOutput	experimentOutputType
1	repr:Experiment_idr0038	repr:ImagingStudy_idr0038	repr:Organism_idr0038	repr:Organism	"Mus musculus"		
2	repr:Experiment_idr0032	repr:ImagingStudy_idr0032	repr:Organism_idr0032	repr:Organism	"Arabidopsis thaliana"		
3	repr:Experiment_idr0020	repr:ImagingStudy_idr0020	repr:Organism_idr0020	repr:Organism	"Homo sapiens"		
4	repr:Experiment_idr0002	repr:ImagingStudy_idr0002	repr:Organism_idr0002	repr:Organism	"Homo sapiens"		
5	repr:Experiment_idr0020	repr:ImagingStudy_idr0020	repr:Library_idr0020_1	repr:Library	"idr0020-screenA-library.txt"		
6	repr:Experiment_idr0002	repr:ImagingStudy_idr0002	repr:Library_idr0002_1	repr:Library	"idr0002-screenA-library.txt"		
7	repr:Experiment_idr0032	repr:SubExperiment_idr0032_1				repr:Assay_idr0032_1	repr:Assay
8	repr:Experiment_idr0032	repr:ImagingStudy_idr0032				repr:Image_idr0032_3126217	repr:Image
9	repr:Experiment_idr0032	repr:ImagingStudy_idr0032				repr:Image_idr0032_3126533	repr:Image
10	repr:Experiment_idr0032	repr:ImagingStudy_idr0032				repr:Image_idr0032_3126118	repr:Image

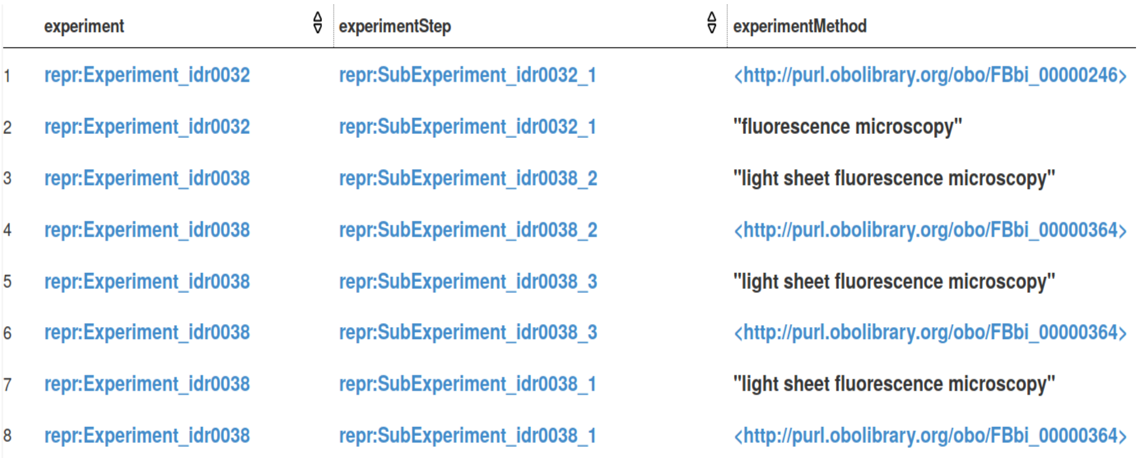
Figure 7.36: A part of SPARQL Query 1 Results

As we could see in the Figure 7.36, each experimentInput and experimentOutput creates separate rows in the results. The query can be split to get either only the input or the output from each step of an experiment. This query can also be used with **FILTER** keyword to extract particular inputs or outputs of a particular type. It also shows how multiple inputs and outputs are connected to a particular step of an experiment. Various variations of this query were used to evaluate the data. For example, (1) List the inputs of type Solution which were used in the bath solution preparation step of the experiments performed by an agent from a particular research group. (2) List the output generated in the first execution of cell 4 of a particular Jupyter Notebook used as a Standard Operating Procedure in a particular experiment which used 'light sheet fluorescence microscopy' method. The SPARQL queries were written for such complex queries and manually compared with the real answers provided by scientists. In the evaluation, the SPARQL queries were answered correctly.

SQ2: Which are the methods and standard operating procedures used?
This query is responsible for getting the methods and the protocols used in an experiment. Figure 7.37 shows the results of this query using the experiments from the IDR repository. As we could see in the results, that some of the experimentMethod is linked to external sources like FBBI[17]. Various complex queries were written taking this as the base query. For example, List all the measurement protocols used in the generation of a particular Image.

```
SELECT DISTINCT * WHERE
{
  ?experiment a repr:Experiment .
  ?experimentStep p-plan:isStepOfPlan ?experiment ;
      repr:usedMethod ?experimentMethod .
}
```

Listing 7.7: SPARQL Query 2



	experiment	experimentStep	experimentMethod
1	repr:Experiment_idr0032	repr:SubExperiment_idr0032_1	<http://purl.obolibrary.org/obo/FBbi_00000246>
2	repr:Experiment_idr0032	repr:SubExperiment_idr0032_1	"fluorescence microscopy"
3	repr:Experiment_idr0038	repr:SubExperiment_idr0038_2	"light sheet fluorescence microscopy"
4	repr:Experiment_idr0038	repr:SubExperiment_idr0038_2	<http://purl.obolibrary.org/obo/FBbi_00000364>
5	repr:Experiment_idr0038	repr:SubExperiment_idr0038_3	"light sheet fluorescence microscopy"
6	repr:Experiment_idr0038	repr:SubExperiment_idr0038_3	<http://purl.obolibrary.org/obo/FBbi_00000364>
7	repr:Experiment_idr0038	repr:SubExperiment_idr0038_1	"light sheet fluorescence microscopy"
8	repr:Experiment_idr0038	repr:SubExperiment_idr0038_1	<http://purl.obolibrary.org/obo/FBbi_00000364>

Figure 7.37: A part of SPARQL Query 2 Results

SQ3: Which are the files and materials that were used in a particular step?
This query is responsible for getting the files and materials used in an experiment. Table 7.5 shows the results of this query using the experiments from the ReceptorLight project. It shows the materials and files used in each step of an experiment.

[17] http://www.ontobee.org/ontology/FBBI

```
SELECT DISTINCT * WHERE {
  ?experiment a :Experiment ; :name ?Experiment .
  {
    ?experiment p-plan:correspondsToVariable ?material .
    ?material a :ExperimentMaterial ;
      :name ?ExperimentMaterial ;
      p-plan:isInputVarOf ?experimentStep ;
      rdfs:label ?MaterialType .
    ?experimentStep rdfs:label ?ExperimentStep
  }
  UNION
 {
    ?plan p-plan:isSubPlanOfPlan ?experiment ; :name ?Plan .
    ?file a :File ; p-plan:isVariableOfPlan ?plan ; :name ?File
 }
}
```

Listing 7.8: SPARQL Query 3

Experiment	**ExperimentMaterial**	**MaterialType**	**ExperimentStep**	**Plan**	**File**
Interaction EGFP-RAD51 and mCherry-RAD52				Calcium phosphate precipitation transfection method	Calcium phosphate precipitation method for cell transfection.pdf
EGFP-RAD51 Time-lapse with Bleomycin	2'-Desoxythimidine	Chemical	Preparation		
EGFP-RAD51 Time-lapse with Bleomycin				Calcium phosphate precipitation transfection method	Calcium phosphate precipitation method for cell transfection.pdf
Staining cellular compartments				Calcium phosphate precipitation transfection method	Calcium phosphate precipitation method for cell transfection.pdf
TK EGFP-RAD51 - CMV Cherry-RAD54	pTK-EGFP-RAD51	Plasmid	Preparation		
TK EGFP-RAD51 - CMV Cherry-RAD54	Bleomycin sulfate	Chemical	Incubation		
TK EGFP-RAD51 - CMV Cherry-RAD54				Calcium phosphate precipitation transfection method	Calcium phosphate precipitation method for cell transfection.pdf
Colocalization of EGFP-RAD51 and Cherry-RAD54	pEGFP-RAD51	Plasmid	Preparation		
Colocalization of EGFP-RAD51 and Cherry-RAD54	pEGFP-RAD51	Plasmid	Description		
Colocalization of EGFP-RAD51 and Cherry-RAD54	pCherry-RAD54	Plasmid	Preparation		
Colocalization of EGFP-RAD51 and Cherry-RAD54	pCherry-RAD54	Plasmid	Transfection		
Colocalization of EGFP-RAD51 and Cherry-RAD54	2'-Desoxythimidine	Chemical	Preparation		
Colocalization of EGFP-RAD51 and Cherry-RAD54				Control of Cell Cycle Distrubution	PI staining FACS_Florian.docx
EGFP-RAD52 / mCherry-RAD54	pCherry-RAD54	Plasmid	Preparation		
EGFP-RAD52 / mCherry-RAD54	pCherry-RAD54	Plasmid	Transfection		
EGFP-RAD52 / mCherry-RAD54	2'-Desoxythimidine	Chemical	Preparation		
EGFP-RAD52 / mCherry-RAD54	Bleomycin sulfate	Chemical	Incubation		
EGFP-RAD52 / mCherry-RAD54				Calcium phosphate precipitation transfection method	Calcium phosphate precipitation method for cell transfection.pdf

Table 7.5: A part of SPARQL Query 3 Results

Various variations of this query were used to evaluate the data. For example, List all the research projects which used both 'pCherry-RAD54' material of type 'Plasmid' and '2'-Desoxythimidine' material of type 'Chemical' and used a particular Jupyter Notebook 'Mean_overlay_analysis.ipynb'.

SQ4: Which are the steps involved in an experiment which used a particular material?

This query is responsible for selectively querying for the steps which used the Plasmid 'pCherry-RAD54'. Table 7.6 shows the results of this query.

```
SELECT DISTINCT * WHERE {
  ?experiment a :Experiment ; :name ?Experiment ;
    p-plan:correspondsToVariable ?material .
  ?material a :ExperimentMaterial ;
    :name ?ExperimentMaterial ;
    p-plan:isInputVarOf ?experimentStep ;
    rdfs:label ?MaterialType FILTER(?ExperimentMaterial='
   pCherry-RAD54') .
  ?experimentStep rdfs:label ?ExperimentStep
}
```

Listing 7.9: SPARQL Query 4

Experiment	ExperimentStep	ExperimentMaterial	MaterialType
Colocalization of EGFP-RAD51 and Cherry-RAD54	Preparation	pCherry-RAD54	Plasmid
Colocalization of EGFP-RAD51 and Cherry-RAD54	Transfection	pCherry-RAD54	Plasmid
EGFP-RAD52 / mCherry-RAD54	Preparation	pCherry-RAD54	Plasmid
EGFP-RAD52 / mCherry-RAD54	Transfection	pCherry-RAD54	Plasmid

Table 7.6: A part of SPARQL Query 4 Results

As we could see from Table 7.6, the Plasmid ‘pCherry-RAD54’ is used in 2 experiments in two different steps.

SQ5: Which are the instruments that are associated with an experiment and their settings when the output was generated?

This query is particularly responsible for the instruments which have been used in an experiment and their settings. Table 7.7 shows the first few results from this query with the experiment, the images and the instruments that generated them.

```
SELECT DISTINCT * WHERE
{
    ?experiment :hasDataset ?dataset ;
      :name ‘EGFP-RAD51 Time-lapse with Bleomycin’ .
    ?dataset prov:hadMember ?image ; :name ?Dataset .
    ?image a :Image ; :name ?Image .
    ?instrument p-plan:correspondsToVariable ?image ;
    repr:hasPart ?instrument_part .
    ?instrument_part repr:hasSetting ?setting ;
      rdf:type ?PartType .
    OPTIONAL { ?setting prov:value ?SettingValue } .
    OPTIONAL { ?instrument_type prov:specializationOf
            ?instrument_part ;
            prov:value ?InstrumentType } .
}
```

Listing 7.10: SPARQL Query 5

Dataset	Image	instrument	instrument_part	PartType	setting
EGFP-RAD51/mCherry-RAD54 time lapse	20180817_RAD51-EGFP_BLM_4h_10min.lif	repr:instrument_1	repr:objective_1	Objective	repr:manufacturer_objective_1
EGFP-RAD51/mCherry-RAD54 time lapse	20180817_RAD51-EGFP_BLM_4h_10min.lif	repr:instrument_1	repr:objective_1	Objective	repr:calibratedmagnification_objective_1
EGFP-RAD51/mCherry-RAD54 time lapse	20180817_RAD51-EGFP_BLM_4h_10min.lif	repr:instrument_1	repr:objective_1	Objective	repr:workingdistance_objective_1
EGFP-RAD51/mCherry-RAD54 time lapse	20180817_RAD51-EGFP_BLM_4h_10min.lif	repr:instrument_1	repr:objective_1	Objective	repr:iris_objective_1
EGFP-RAD51/mCherry-RAD54 time lapse	20180817_RAD51-EGFP_BLM_4h_10min.lif	repr:instrument_1	repr:objective_1	Objective	repr:immersion_7
EGFP-RAD51/mCherry-RAD54 time lapse	20180817_RAD51-EGFP_BLM_4h_10min.lif	repr:instrument_1	repr:objective_1	Objective	repr:model_objective_1
EGFP-RAD51/mCherry-RAD54 time lapse	20180817_RAD51-EGFP_BLM_4h_10min.lif	repr:instrument_1	repr:objective_1	Objective	repr:serialnumber_objective_1
EGFP-RAD51/mCherry-RAD54 time lapse	20180817_RAD51-EGFP_BLM_4h_10min.lif	repr:instrument_1	repr:objective_1	Objective	repr:objectivesettings_1
EGFP-RAD51/mCherry-RAD54 time lapse	20180817_RAD51-EGFP_BLM_4h_10min.lif	repr:instrument_1	repr:objective_1	Objective	repr:nominalmagnification_objective_1
EGFP-RAD51/mCherry-RAD54 time lapse	20180817_RAD51-EGFP_BLM_4h_10min.lif	repr:instrument_1	repr:objective_1	Objective	repr:lotnumber_objective_1

Table 7.7: A part of SPARQL Query 5 Results

Each experiment has several images as output. Each image is generated by an instrument. The instrument has several parts. Each part has several settings. In the Table 7.7, we could see the settings from the Objective of the Microscope which generated the image ‘20180817_RAD51-EGFP_BLM_4h_10min.lif’. Many instruments and their parts are associated with a particular image. The settings need to be queried seperately to get all the configurations made in each of the instruments' parts.

SQ6: Which are the agents directly or indirectly responsible for an experiment?

This query is responsible for getting all the agents and their role in an experiment. Figure 7.38 shows the result from this query.

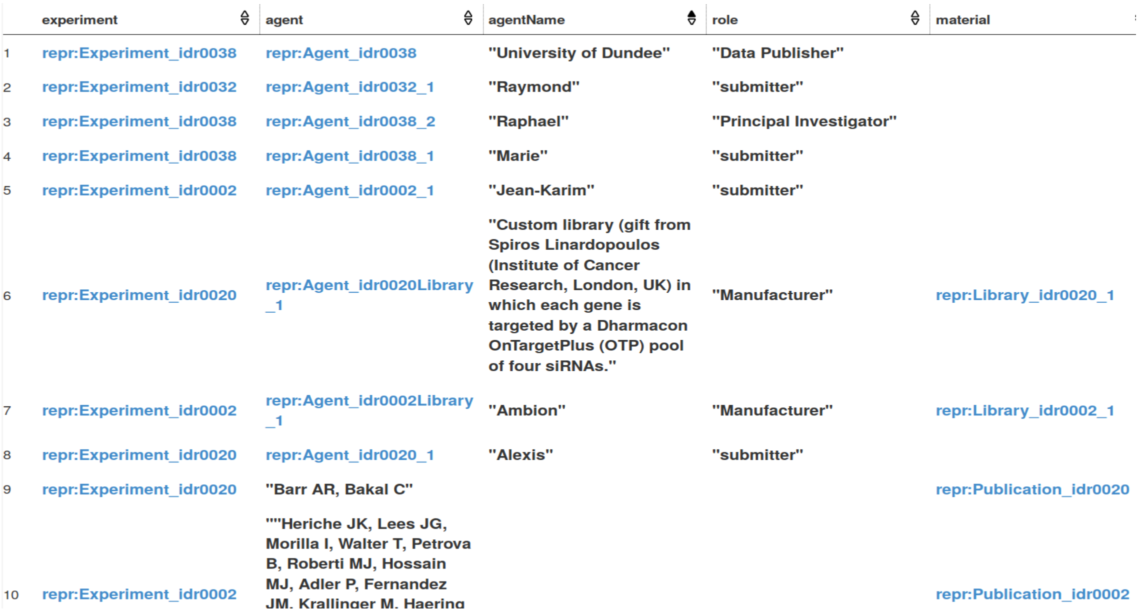
```
SELECT DISTINCT ?experiment ?agent ?agentName ?role ?material
   WHERE
{
    ?experiment a repr:Experiment .
    ?step p-plan:isStepOfPlan ?experiment .
    {
        ?experiment prov:wasAttributedTo ?agent.
        OPTIONAL { ?agent foaf:givenName ?agentName }
        ?agent repr:hasRole ?role .
    }
    UNION
    {
        ?material p-plan:isInputVarOf ?step .
        ?material prov:wasAttributedTo ?agent .
        OPTIONAL {
            ?agent repr:name ?agentName ; repr:hasRole ?role .
        }
    }
    UNION
    {
        ?material p-plan:isOutputVarOf ?step .
        ?material prov:wasAttributedTo ?agent .
        OPTIONAL {
            ?agent repr:name ?agentName ; repr:hasRole ?role .
        }
    }
}
```

Listing 7.11: SPARQL Query 6

From the results, we see that the agents who are directly responsible (Experimenter, Principal Investigator) for an experiment are added by the scientists. In addition to that, the agents who are indirectly responsible for an experiment like the Manufacturer, Distributor, Data Publisher are also considered important by scientists and linked to an experiment. The survey results also showed that it is important to share the name, contacts, and roles of the agents directly or indirectly involved in an experiment (see Figure 7.16).

SQ7: Who created this experiment and when? When was the experiment started?

This query is responsible for getting all the agents and temporal aspects of an experiment. Table 7.8 shows the result from this query.



	experiment	agent	agentName	role	material
1	repr:Experiment_idr0038	repr:Agent_idr0038	"University of Dundee"	"Data Publisher"	
2	repr:Experiment_idr0032	repr:Agent_idr0032_1	"Raymond"	"submitter"	
3	repr:Experiment_idr0038	repr:Agent_idr0038_2	"Raphael"	"Principal Investigator"	
4	repr:Experiment_idr0038	repr:Agent_idr0038_1	"Marie"	"submitter"	
5	repr:Experiment_idr0002	repr:Agent_idr0002_1	"Jean-Karim"	"submitter"	
6	repr:Experiment_idr0020	repr:Agent_idr0020Library_1	"Custom library (gift from Spiros Linardopoulos (Institute of Cancer Research, London, UK) in which each gene is targeted by a Dharmacon OnTargetPlus (OTP) pool of four siRNAs."	"Manufacturer"	repr:Library_idr0020_1
7	repr:Experiment_idr0002	repr:Agent_idr0002Library_1	"Ambion"	"Manufacturer"	repr:Library_idr0002_1
8	repr:Experiment_idr0020	repr:Agent_idr0020_1	"Alexis"	"submitter"	
9	repr:Experiment_idr0020	"Barr AR, Bakal C"			repr:Publication_idr0020
10	repr:Experiment_idr0002	""Heriche JK, Lees JG, Morilla I, Walter T, Petrova B, Roberti MJ, Hossain MJ, Adler P, Fernandez JM, Krallinger M, Haering			repr:Publication_idr0002

Figure 7.38: A part of SPARQL Query 6 Results

```
SELECT DISTINCT * WHERE
{
    ?experiment a repr:Experiment ;
                repr:name ?Experiment ;
                prov:wasAttributedTo ?Agent .
    OPTIONAL {
        ?experiment prov:startedAtTime ?startedAtTime ;
                    prov:generatedAtTime ?generatedAtTime
    }
}
```

Listing 7.12: SPARQL Query 7

Experiment	Agent	startedAtTime	generatedAtTime
EGFP-RAD51/mCherry-RAD54	repr:Researchgroup_Experiment_2	2018-09-30T10:11:00	2019-03-01T21:07:07+01:00
EGFP-RAD51/mCherry-RAD54	repr:ContactPerson_Experiment_2	2018-09-30T10:11:00	2019-03-01T21:07:07+01:00
EGFP-RAD51/mCherry-RAD54	repr:Project_Experiment_2	2018-09-30T10:11:00	2019-03-01T21:07:07+01:00
EGFP-RAD51/mCherry-RAD54	repr:ExperimenterGroup_53	2018-09-30T10:11:00	2019-03-01T21:07:07+01:00

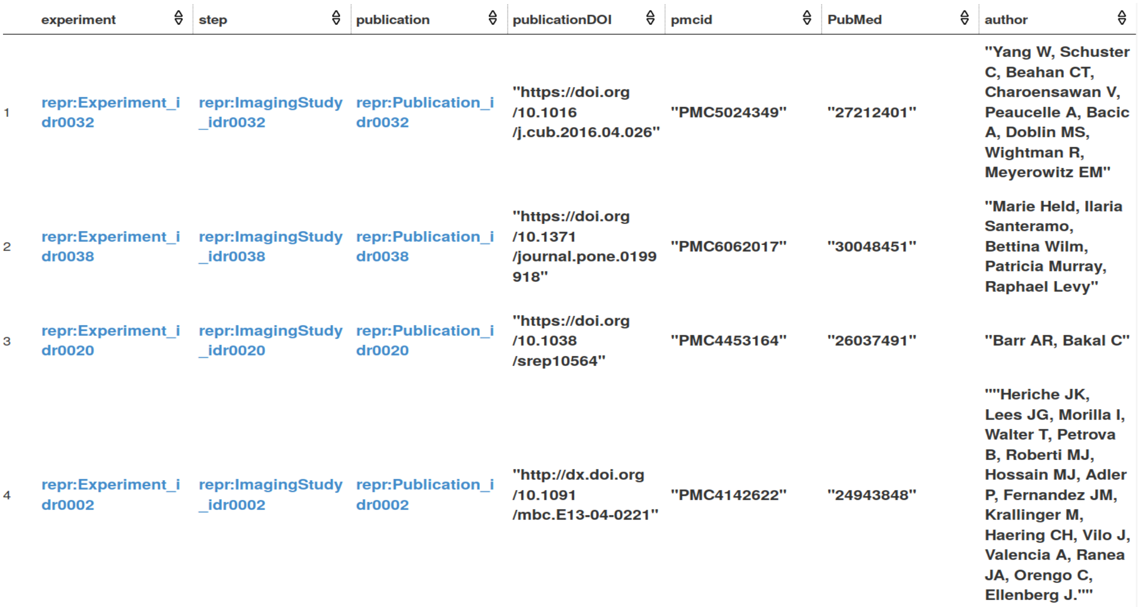
Table 7.8: A part of SPARQL Query 7 Results

SQ8: Which are the publications or external resources that were referenced?

This query is responsible for the publications that resulted for an experiment or external resources that were referenced. Figure 7.39 shows the result from this query.

```
SELECT DISTINCT * WHERE
{
    ?experiment a repr:Experiment .
    ?step p-plan:isStepOfPlan ?experiment .
    ?publication p-plan:isOutputVarOf ?step ;
                rdf:type repr:Publication ;
                repr:doi ?publicationDOI .
    OPTIONAL { ?publication repr:pmcid ?pmcid  } .
    OPTIONAL { ?publication repr:pubmedid ?PubMed }
    OPTIONAL { ?publication prov:wasAttributedTo ?author }
}
```

Listing 7.13: SPARQL Query 8



	experiment	step	publication	publicationDOI	pmcid	PubMed	author
1	repr:Experiment_idr0032	repr:ImagingStudy_idr0032	repr:Publication_idr0032	"https://doi.org/10.1016/j.cub.2016.04.026"	"PMC5024349"	"27212401"	"Yang W, Schuster C, Beahan CT, Charoensawan V, Peaucelle A, Bacic A, Doblin MS, Wightman R, Meyerowitz EM"
2	repr:Experiment_idr0038	repr:ImagingStudy_idr0038	repr:Publication_idr0038	"https://doi.org/10.1371/journal.pone.0199918"	"PMC6062017"	"30048451"	"Marie Held, Ilaria Santeramo, Bettina Wilm, Patricia Murray, Raphael Levy"
3	repr:Experiment_idr0020	repr:ImagingStudy_idr0020	repr:Publication_idr0020	"https://doi.org/10.1038/srep10564"	"PMC4453164"	"26037491"	"Barr AR, Bakal C"
4	repr:Experiment_idr0002	repr:ImagingStudy_idr0002	repr:Publication_idr0002	"http://dx.doi.org/10.1091/mbc.E13-04-0221"	"PMC4142622"	"24943848"	""Heriche JK, Lees JG, Morilla I, Walter T, Petrova B, Roberti MJ, Hossain MJ, Adler P, Fernandez JM, Krallinger M, Haering CH, Vilo J, Valencia A, Ranea JA, Orengo C, Ellenberg J.""

Figure 7.39: A part of SPARQL Query 8 Results

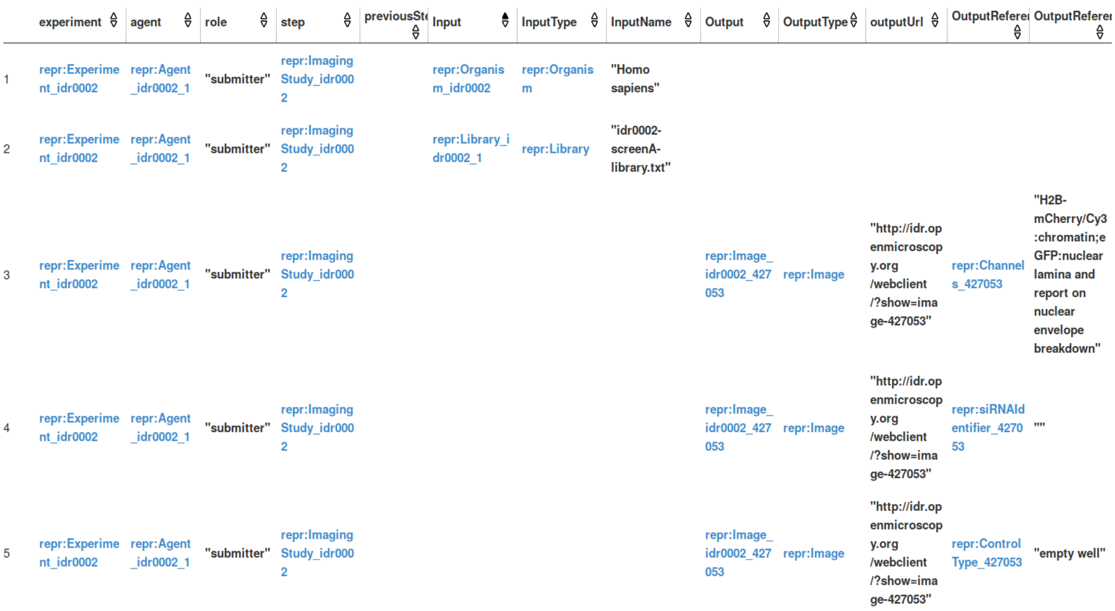
SQ9: What is the complete path taken by a scientist for an experiment?

This query is responsible for getting the complete path for an experiment.

```
SELECT DISTINCT * WHERE
{
    ?experiment a repr:Experiment ;
      prov:wasAttributedTo ?agent ; repr:hasDataset ?dataset ;
      prov:generatedAtTime ?generatedAtTime .
    ?agent repr:hasRole ?role .
    ?dataset prov:hadMember ?image .
    ?instrument p-plan:correspondsToVariable ?image ;
      repr:hasPart ?instrument_part .
    ?instrument_part repr:hasSetting ?setting .
    ?plan p-plan:isSubPlanOfPlan ?experiment .
    ?variable p-plan:isVariableOfPlan ?plan .
    ?step p-plan:isStepOfPlan ?experiment .
    OPTIONAL { ?step p-plan:isPrecededBy ?previousStep } .
    {
       ?Input p-plan:isInputVarOf ?step ; rdf:type ?InputType .
       OPTIONAL { ?Input repr:name ?InputName } .
    }
    UNION {
       ?Output p-plan:isOutputVarOf ?step ;
        rdf:type ?OutputType .
       OPTIONAL { ?Output repr:name ?OutputName } .
       OPTIONAL { ?Output repr:isAvailableAt ?outputUrl } .
```

```
        OPTIONAL { ?Output repr:reference ?OutputReference .
           ?OutputReference rdf:value ?OutputReferenceValue
        }
    }
}
```

Listing 7.14: SPARQL Query 9



	experiment	agent	role	step	previousSt	Input	InputType	InputName	Output	OutputType	outputUrl	OutputRefere	OutputRefere
1	repr:Experiment_idr0002	repr:Agent_idr0002_1	"submitter"	repr:ImagingStudy_idr0002		repr:Organism_idr0002	repr:Organism	"Homo sapiens"					
2	repr:Experiment_idr0002	repr:Agent_idr0002_1	"submitter"	repr:ImagingStudy_idr0002		repr:Library_idr0002_1	repr:Library	"idr0002-screenA-library.txt"					
3	repr:Experiment_idr0002	repr:Agent_idr0002_1	"submitter"	repr:ImagingStudy_idr0002					repr:Image_idr0002_427053	repr:Image	"http://idr.openmicroscopy.org/webclient/?show=image-427053"	repr:Channels_427053	"H2B-mCherry/Cy3:chromatin;eGFP:nuclear lamina and report on nuclear envelope breakdown"
4	repr:Experiment_idr0002	repr:Agent_idr0002_1	"submitter"	repr:ImagingStudy_idr0002					repr:Image_idr0002_427053	repr:Image	"http://idr.openmicroscopy.org/webclient/?show=image-427053"	repr:siRNAIdentifier_427053	""
5	repr:Experiment_idr0002	repr:Agent_idr0002_1	"submitter"	repr:ImagingStudy_idr0002					repr:Image_idr0002_427053	repr:Image	"http://idr.openmicroscopy.org/webclient/?show=image-427053"	repr:ControlType_427053	"empty well"

Figure 7.40: A part of SPARQL Query 9 Results

The important elements required for the reproducibility of a scientific experiment are used to describe its complete path. Here, we queried the experiment with its associated agents and their role, the plans and steps involved, the input and output of each step, the order of steps, and the instruments and their setting. This query can be further expanded by querying for additional information like the materials, publications, external resources, methods, etc. used in each step of an experiment. Figure 7.40 shows the part of the result for a particular experiment called 'Focused mitotic chromsome condensaton screen using HeLa cells'. The results show that this query helps in getting all the important elements required for reproducibility. We also see that all these elements now are linked together to describe the complete path. The experiment is linked to the computational and non-computational steps. It is possible to query for all the elements mentioned in the REPRODUCE-ME Data Model (see Section 4.4).

We also observed that for certain experiments which did not provide the complete data for some elements, the query returned null. So the query needs to be tweaked to include the OPTIONAL keyword to get the results from the query. Another thing that we notice during the evaluation is that the results are spread across several rows in the table. In the Dashboard, when we show these results, the filter option provided in the table helps the user to search for particular columns.

SQ10: List all the experiments which use growth protocol (EFO_0003789) and studies on "Homo sapiens" and resulted in phenotype "shorter prophase" which passed the quality control.

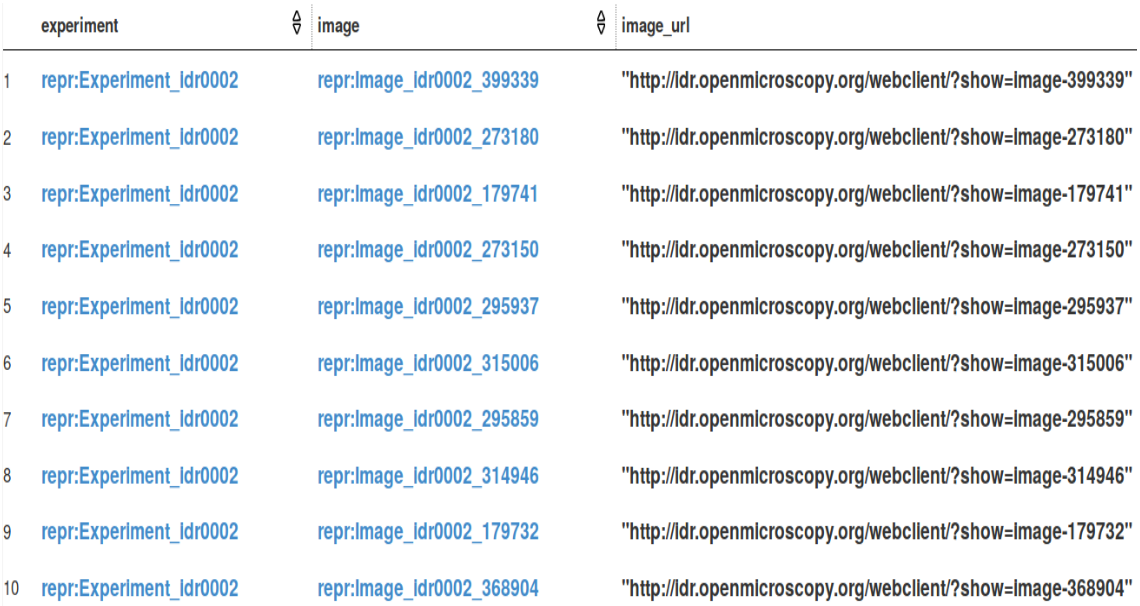
This query is responsible for selectively querying for complex questions according to the requirements of a scientist. Figure 7.41 shows the result of this query.

```
SELECT DISTINCT ?experiment ?image ?image_url WHERE
{
  ?experiment a repr:Experiment .
  ?step p-plan:isStepOfPlan ?experiment .
  ?image p-plan:isOutputVarOf ?step ;
    a repr:Image .
  ?screen p-plan:isOutputVarOf ?step ;
    a repr:Screen ;
    p-plan:isVariableOfPlan ?protocol .
  ?protocol a repr:Protocol ;
    repr:type ?type FILTER(?type="EFO_0003789") .
  ?image repr:reference ?organism .
  ?organism a repr:Organism ;
    rdf:value ?organismvalue Filter(?organismvalue="Homo
   sapiens") .
  ?image repr:reference ?phenotype .
  ?phenotype a repr:Phenotype ;
    rdf:value ?value Filter(?value="shorter prophase") .
  ?image repr:reference ?QualityControl .
  ?QualityControl a repr:QualityControl ;
    rdf:value ?QualityControlValue FILTER(?QualityControlValue
   ="pass") .
  ?image repr:isAvailableAt ?image_url
}
```

Listing 7.15: SPARQL Query 10

7.5.1 Discussion

In this evaluation, we focused on answering the competency questions which were defined for the development of the REPRODUCE-ME ontology. The ontology was also evaluated by using it in CAESAR to describe the experiments related to microscopy. The competency questions were translated to SPARQL queries by computer scientists. This is because scientists from life sciences are not aware of writing SPARQL queries. However, to overcome this limitation, we provide them with the visualization of the results from these SPARQL queries through Dashboard and ProvTrack. The SPARQL queries of the competency questions (**CQ1-CQ24**) were answered by the REPRODUCE-ME ontology. The results of SPARQL queries were manually compared using Dashboard and ProvTrack. Their correctness was evaluated by the



	experiment	image	image_url
1	repr:Experiment_idr0002	repr:Image_idr0002_399339	"http://idr.openmicroscopy.org/webclient/?show=image-399339"
2	repr:Experiment_idr0002	repr:Image_idr0002_273180	"http://idr.openmicroscopy.org/webclient/?show=image-273180"
3	repr:Experiment_idr0002	repr:Image_idr0002_179741	"http://idr.openmicroscopy.org/webclient/?show=image-179741"
4	repr:Experiment_idr0002	repr:Image_idr0002_273150	"http://idr.openmicroscopy.org/webclient/?show=image-273150"
5	repr:Experiment_idr0002	repr:Image_idr0002_295937	"http://idr.openmicroscopy.org/webclient/?show=image-295937"
6	repr:Experiment_idr0002	repr:Image_idr0002_315006	"http://idr.openmicroscopy.org/webclient/?show=image-315006"
7	repr:Experiment_idr0002	repr:Image_idr0002_295859	"http://idr.openmicroscopy.org/webclient/?show=image-295859"
8	repr:Experiment_idr0002	repr:Image_idr0002_314946	"http://idr.openmicroscopy.org/webclient/?show=image-314946"
9	repr:Experiment_idr0002	repr:Image_idr0002_179732	"http://idr.openmicroscopy.org/webclient/?show=image-179732"
10	repr:Experiment_idr0002	repr:Image_idr0002_368904	"http://idr.openmicroscopy.org/webclient/?show=image-368904"

Figure 7.41: A part of SPARQL Query 10 Results

domain experts [Samuel et al., 2018]. The elements described in the REPRODUCE-ME Data model required for reproducibility are linked together to an experiment to describe its complete path. Each of the competency questions addressed the different elements of the REPRODUCE-ME Data Model. We also evaluated the ontology with different variations in the competency questions. With the help of SPARQL queries, we saw that some experiments had missing provenance data on time, settings, etc. We also observe that the output of the query for finding the complete path of scientific experiment results in several rows in the table. Hence, in some cases where the experiment has several inputs and outputs with several executions, the response time can exceed the normal query response time and result in server error from the SPARQL endpoint. To avoid this issue, we split the queries and combine their results together in ProvTrack. We also group the entities, agents, activities, steps, and plans in ProvTrack to help users visualize the complete path of an experiment.

7.5.2 Visualization

To evaluate the main hypothesis of our work, it is important that the provenance of scientific experiments is visualized to the scientists in an appealing manner since everyone is not aware of writing SPARQL queries. In Section 7.5, we saw that the competency questions are answered using the REPRODUCE-ME ontology. The visualization modules in CAESAR provide users the way to see the results from these competency questions. To do so, the Dashboard provides the provenance of scientific experiments conducted in a project with tabular design, while ProvTrack provides the visualization of the provenance graph of each scientific experiment. To evaluate the modules, we used the experiments from our knowledge base. First, we conducted a performance evaluation of each of the modules. In order to increase the performance and usability of the dashboard, we designed the dashboard with lazy load. Each panel makes an asynchronous call to the backend and is rendered as soon as the response is back without waiting for all the responses to arrive. In ProvTrack, we query the provenance of each scientific experiment and combine the results from the competency questions to visualize the complete path.

We performed a user-based evaluation of CAESAR. 7 participants were invited for the survey, of which 6 participants responded to the questions. The participants of this evaluation were the scientists of ReceptorLight project who use CAESAR in their daily work. In addition to them, there were other biology students, who closely work with microscopy images and are not part of ReceptorLight project, participated in this evaluation. The scientists from ReceptorLight project were given training on CAESAR and its workflow on documenting experimental data. Apart from the internal meetings, the trainings were done throughout the years from 2016-2018 (17.06.2016, 19.07.2016, 07.06.2017, 09.04.2018 and 16.06.2018). As part of these trainings, scientists were asked to upload their experimental data to CAESAR. Table 7.4 shows the statistics of the experimental data that were uploaded as part of this process. A part of this data was used for the evaluation. The purpose of this study was to see how the users find CAESAR useful with respect to the features it provides. The questionnaire along with the responses are available in Appendix D.1. None of the questions in the study was mandatory. Figures 7.42, 7.43 and 7.44 show the results from the user evaluation of CAESAR.

In the first section of the study, we asked how the features in CAESAR help in improving their daily research work. All the participants either strongly agreed or

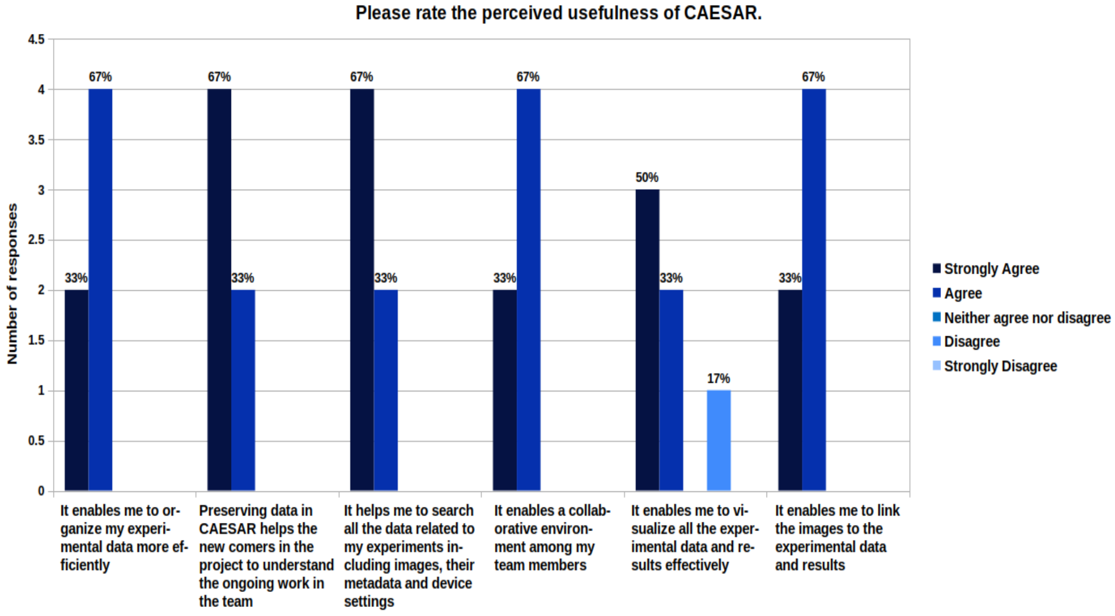


Figure 7.42: CAESAR User Evaluation: The perceived usefulness of CAESAR

agreed that CAESAR enables them to organize their experimental data efficiently, preserve data for the newcomers, search all the data, provide a collaborative environment and link the experimental data with results as seen in Figure 7.42. **83%** of the participants either strongly agreed or agreed that it helps to visualize all the experimental data and results effectively, while **17%** of them disagreed on that.

In the next section (Figure 7.43), we asked on the perceived usefulness of CAESAR.

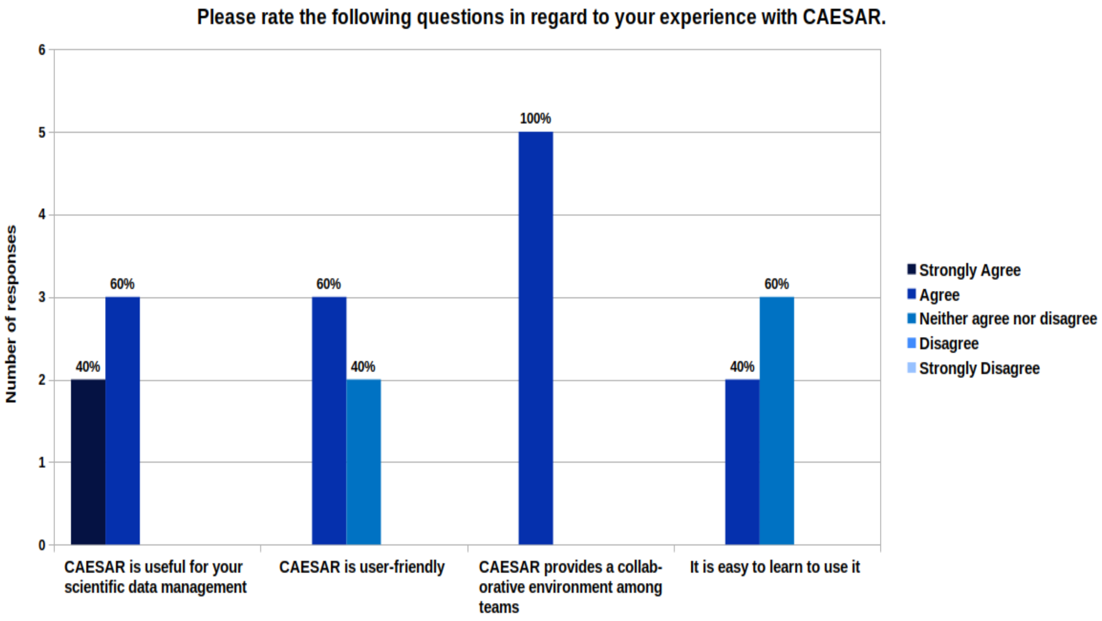


Figure 7.43: CAESAR User Evaluation: Experience with CAESAR

60% of the users consider CAESAR user-friendly while **40%** of them had a neutral response. **40%** of the participants agreed that CAESAR is easy to learn to use and **60%** had a neutral response. The response for this was mentioned that CAESAR provides a lot of features and they found it little difficult to follow. However, all the participants strongly agreed or agreed that CAESAR is useful for scientific data management and provides a collaborative environment among teams.

In the last section, we evaluated each feature provided by CAESAR by focusing on the important visualization modules. ProvTrack was strongly liked or liked by all the participants as seen in Figure 7.44. For the Dashboard, **80%** of them either strongly liked or liked, while **20%** of them had a neutral response. **60%** of the users strongly liked or liked ProvBook, while other **40%** had a neutral response. The reasons for the neutral response was because they were new to scripting.

We also asked to provide the overall feedback of CAESAR along with its positive aspects and the things to improve. We got 3 responses to this question which are available in Appendix D.1.

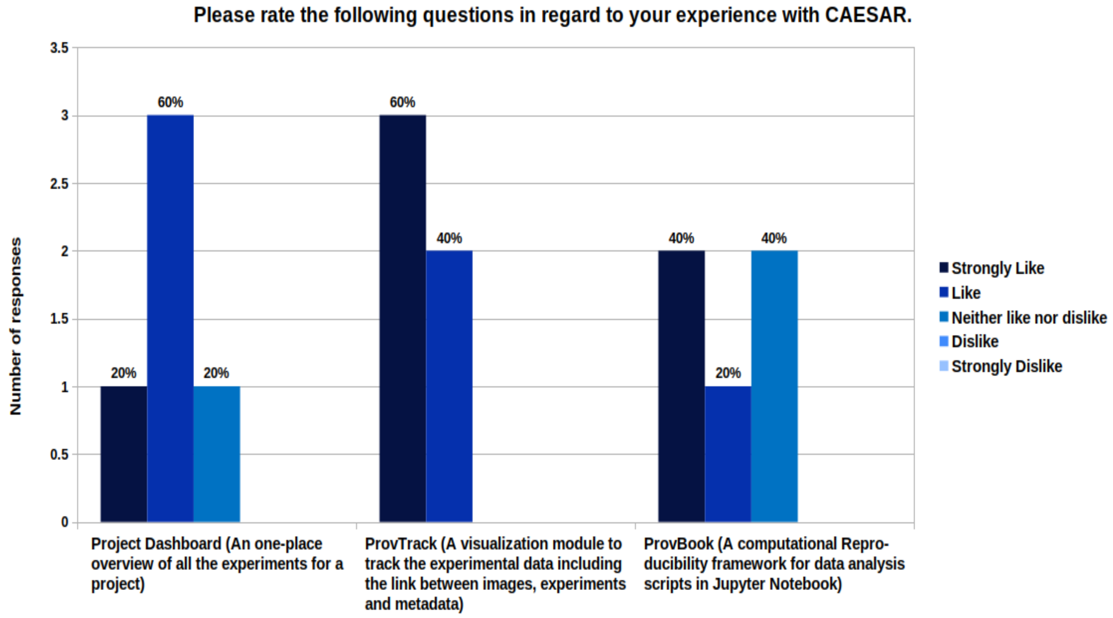


Figure 7.44: CAESAR User Evaluation: The features of CAESAR

7.5.3 Discussion

In our user study of CAESAR, we targeted both the regular users and the users who are new to the system. Even though we had a small group of participants, they either agreed or liked the features provided by it. A strong agreement was seen among the participants that it helps to preserve data for the newcomers to understand the ongoing work in the team. The survey results in Section 7.3 had shown that newcomers face difficulty in finding, accessing and reusing data in a team (see Figure 7.9). Hence, we could see that CAESAR addresses this issue for the newcomers. This understanding of the ongoing work in the team comes from the linking of experimental data and results. This is achieved using the visualization of the overall view of the experimental data. The results from the study show that among the two visualization modules, ProvTrack was preferred over Dashboard by scientists. Even though both serve different purposes (Dashboard for an overall view of the experiments conducted in a Project and the ProvTrack for backtracking the results of one experiment), the users preferred the provenance graph to be visualized with detailed information on clicking. The participants did not have the knowledge of Semantic Web technologies and were also not familiar with writing SPARQL queries. Hence, we did not perform any user study on writing SPARQL queries to answer competency questions. In our surveys, we did not use any technical terms like provenance, ontology, Semantic Web, etc. This was to make sure that all the questions can be answered by participants even without knowing the computer science technical terms. The survey shows that the visualization of the experimental data and results using ProvTrack supported by the REPRODUCE-ME ontology helps the scientists without worrying about the underlying technologies. Hence, it validates our main hypothesis.

7.6 Summary

This chapter presented the evaluation of our research work. Table 7.9 provides the summary of the evaluation of this thesis work. The evaluation of each module was done by different users using real-life experiments. The first user interviews presented some important points that need to be addressed for reproducibility and data management process. The next section presented a user-based survey on the understanding of experiments and research practices. The survey results showed that there is a reproducibility crisis. Several factors that lead to poor reproducibility of results according to users' experience were shared. The results also showed that sharing data and results are not just enough for reproducibility. The methods, negative and intermediate results, steps and execution environment along with settings have also got high priority in sharing experimental metadata. We evaluated the support of computational reproducibility using ProvBook. It was used in different scenarios by different users with a different set of Jupyter Notebooks. The results showed how ProvBook helped to track the provenance of computational experiments hence helped in supporting reproducibility. In the last section, we evaluated the modules in CAESAR. The REPRODUCE-ME ontology was evaluated by using in CAESAR using the data uploaded by the scientists from the CRC ReceptorLight project and IDR repository. We showed how each competency question was answered using the ontology. The user-based survey showed that CAESAR is useful for provenance data management especially the visualization modules. However, the performance time needs room for improvement. Since CAESAR provides a rich set of features, training is required for the scientists to be familiarized with the system.

Experiment	**Hypothesis, Goal, Requirement**	**Outcome/Remarks**
The REPRODUCE-ME Data Model Survey on understanding experiments and research practices for reproducibility	H1 & Goal1	Reproducibility Crisis exists. Each element in the REPRODUCE-ME Data Model is important for reproducibility.
Computational Reproducibility Computational experiments performed by different users in different environments	H3 Goal2	Intermediate, negative, and final results from different users are tracked by ProvBook.
The input, output, execution time and the order in different executions of a computational experiment	H3 & Goal2	Supported by the provenance captured by ProvBook
Provenance difference of the results of computational steps	H4 & Goal2	Comparison of results with the results from the original author is supported by ProvBook.
Performance of ProvBook with respect to time	Goal2	Difference in the execution time of cells with and without ProvBook is negligible.
Performance of ProvBook with respect to space	Goal2	Provenance data grows more than actual data
Environmental attributes of the execution of computational experiments	H1.2 & Goal2	Supported by the provenance captured by ProvBook
Competency Questions **CQ11-CQ24**	H1, H2 & Goal1	Answered using the REPRODUCE-ME ontology
CAESAR Competency Questions **CQ1-CQ10**	H1, H2 & Goal1	Answered using the REPRODUCE-ME ontology
User Evaluation	H5 & Goal3	Interactive provenance graph provided by ProvTrack provides a complete path of an experiment and is useful.
ProvBook	R1, R2, R3, R5, R6 & R7	Supported
CAESAR	R1, R2, R3, R4, R5, R6 & R7	Supported

Table 7.9: Summary of the evaluation of our research work. The table presents the experiments which validate the hypothesis and their outcome.

Chapter 8

Conclusions and Future Work

This chapter concludes this dissertation. In Section 8.1, we provide a brief summary of this research work. Section 8.2 brings the dissertation to a close by reviewing its contributions and the extent to which the hypotheses and goals are achieved. In Section 8.3, we examine the future lines of work.

8.1 Summary

The research problem that we addressed in this thesis was how to support understandability, reproducibility, and reuse of scientific experiments. The motivation behind our work comes from the scientists who want to understand, reproduce and reuse each others' results in a collaborative research environment. As our first step, we studied the different practices and requirements of scientists in life sciences in the context of scientific data management for reproducibility. Every research group had its own way of storing experimental data using different techniques and tools. However, several challenges were faced by scientists in tracking the provenance of results. The lack of a link between the data and results from the computational and non-computational steps of an experiment is one of the main challenges that concerned reproducibility of results.

We focused our research work based on three key questions: **(1)** How to describe and represent the complete path of a scientific experiment? **(2)** How to support computational reproducibility? **(3)** How to develop a collaborative framework which provides access to the complete path of a scientific experiment? Three main contributions emerged to answer these three questions. We also combined the first and second contributions together in the third contribution to provide scientists a single place to visualize the complete path of a scientific experiment.

The three main contributions are as follows: **(1)** We developed the REPRODUCE-ME Data Model and ontology to describe the complete path of a scientific experiment consisting of results from the computational and non-computational steps using semantic web technologies. **(2)** To support computational reproducibility, we

developed ProvBook to capture, manage, compare and visualize the provenance information of computational notebooks. **(3)** We were able to develop an end-to-end provenance management platform, CAESAR, to help scientists working collaboratively.
Each contribution was used in real-world scenarios and evaluated. Based on the evaluation results, we conclude the following: **(1)** Reproducibility is an important concern in data-intensive science and needs much attention to improve the current situation. **(2)** The REPRODUCE-ME data model describes the important components required for reproducibility of scientific experiments and it can be extended to meet the requirements for each scientific field. **(3)** ProvBook, an easy-to-use tool which provides the support to capture, represent, store, compare and visualize provenance is an example to support computational reproducibility. **(4)** CAESAR which provides support for the end-to-end provenance management from the beginning of an experiment to its end addresses the major concerns of the scientists in the context of understandability, reproducibility, and reuse. The results of this thesis with all the information of the contributions are available online[1].

8.2 Contributions

In this section, we summarize our contributions with regard to the research problems, goals, and hypothesis that we defined in Chapter 2. Table 8.1 provides the summary of the results of this thesis work.

8.2.1 The REPRODUCE-ME Data Model and the ontology

We first investigated the different possibilities to describe the complete path of a scientific experiment. This study led us to the benefits of Semantic Web technologies and linked data in supporting understandability and sharing of domain knowledge. Based on our understanding of different experimental workflows from various scientists, we precisely defined reproducibility and repeatability. In the next step, we did a literature survey on different data models which describe provenance information in general. Inspired by the W7, PRIMAD, and PROV-DM models, we developed the REPRODUCE-ME data model. We described eight main components required for the reproducibility of experiments: Data, Agent, Activity, Plan, Step, Setting, Instrument, Material. We defined each component and further provided their classifications. To encode the REPRODUCE-ME data model in OWL, we reviewed whether the existing provenance models are adequate enough for capturing provenance information of the complete path of a scientific experiment. Based on the review, we selected the W3C recommendation PROV Ontology (PROV-O)

[1] `https://w3id.org/reproduceme/research`

Hypothesis	Goals	Contribution	Remarks
H1	Goal1	REPRODUCE-ME DM	A conceptual model is developed to describe a complete path of a scientific experiment.
H1.1	Goal1	REPRODUCE-ME DM	The data model is developed to represent the relationship between the data, the steps and the results of a non-computational experiment.
H1.2	Goal1	REPRODUCE-ME DM	The data model is developed to represent the relationship between the data, the steps and the results of a computational experiment.
H1.2	Goal1	REPRODUCE-ME DM	The data model is developed to represent the relationship between the computational and non-computational aspects of a scientific experiment.
H2	Goal1	REPRODUCE-ME Ontology	Semantic Web technologies are used by extending the existing standards to answer the defined competency questions.
H3	Goal2	ProvBook	Computational notebooks are extended to provide provenance support for reproducibility.
H4	Goal2	ProvBook	Demonstrated the support of computational reproducibility using ProvBook in different use-case scenarios.
H5	Goal3	CAESAR	Demonstrated the use of the collaborative framework to capture, represent and visualize a complete path of a scientific experiment.

Table 8.1: Summary of the contributions along with hypothesis and goals

which describes the entities, agents, and activities of an application system. We used PROV-O as a foundation to represent provenance information of a scientific experiment and extended it to meet the domain requirements. To provide more details to the input and output data used and generated in each step of an experiment, we used P-Plan, which is primarily used to describe abstract scientific workflows and their executions. We developed the REPRODUCE-ME ontology by extending PROV-O and P-Plan to describe the provenance information of scientific experiments. The concepts and properties in REPRODUCE-ME were aligned with PROV-O and P-Plan. The REPRODUCE-ME ontology not only provides the description of non-computational steps but also the computational steps from the execution of computational notebooks and scripts. Then, it interlinks the data, the results, and the execution environment of these steps to describe the complete path of an experiment. We evaluated this approach by applying it in experiments related to high-end light microscopy. The competency questions answered by the REPRODUCE-ME ontology addressed different aspects of provenance information of a scientific experiment. The user-based survey also showed us that the components defined in the REPRODUCE-ME data model are important for reproducibility of results. This contribution helped to achieve Goal1 by validating the hypothesis H1 and H2.

8.2.2 Support of Computational Reproducibility

The importance of data science and computational research for scientists in their daily work drew our attention to computational reproducibility. Therefore, we first studied different computational tools in the context of scientific workflows, scripts, and computational notebooks. We focused our research on computational notebooks because of their wide adoption and how they help in reproducibility by sharing code along with the results and documentation. Based on our study, we understood that there is however limited provenance support in these notebooks. Their provenance support was limited because there was no approach to track and compare the results of the different executions. To provide provenance support, we developed ProvBook (Provenance of the Notebook) which is an extension of Jupyter Notebooks. It captures and visualizes the provenance information of different executions of the cells in the notebook over the course of time. It also provides the user the facility to see the difference between the results from the original experimenter with the current ones. This feature can also be used in tracking the intermediate and negative results. In addition to that, ProvBook allows the user to download and share the notebook along with its provenance in Resource Description Framework (RDF) described using the REPRODUCE-ME ontology. This shared RDF can also be converted back to an executable notebook. Besides computational notebooks, we also provide a semantic representation of provenance of scripts and their executions using the REPRODUCE-ME ontology. One of the benefits of representing the provenance information of computational notebooks and scripts is that they can be combined with the metadata of the experiments which used them. In this way, we track the complete path of scientific experimental results using the REPRODUCE-ME ontology. We used publicly available Jupyter Notebooks collected from GitHub and evaluated ProvBook. We used them in different cases by changing variables according to Reproducibility and Repeatability Matrix (Table 4.1 and 4.2). The results demonstrated that ProvBook supports computational reproducibility. The simplicity of this tool is another highlighted feature. This contribution helped to achieve Goal2 by validating the hypothesis H3 and H4.

8.2.3 CAESAR

We introduced the notion of "end-to-end provenance management" of scientific experiments in order to support reproducibility. This led to the development of CAESAR (**C**oll**A**borative **E**nvironment for **S**cientific **A**nalysis with **R**eproducibility) which provides scientific data management. We narrowed our scope to the provenance management of experiments in life-sciences particularly concerned with imaging datasets. Based on the literature survey on the existing tools, we selected OMERO which provides rich features for the management of images. We extended OMERO to provide end-to-end management of provenance of scientific experiments. To do so, we developed modules to capture, represent, store, and visualize provenance. The provenance capture module provides a form-based metadata editor with rich facilities. The metadata extracted from the images is also linked to the experimental data provided by scientists. JupyterHub and ProvBook installed in CAESAR provide a collaborative computational environment and capture provenance of computational tasks. We use the REPRODUCE-ME ontology to describe and integrate the provenance information collected from different sources. The different sources include the metadata added by scientists through metadata editor, automatic extraction of settings and execution environment collected from images generated from different instruments, and the provenance information of the computational steps collected using ProvBook. This information from various sources is linked and stored using ontology-based data access. The federation store of multiple databases in CAESAR and the mapping of the underlying databases to the ontology provide a semantic approach to query the data using the rdf4j SPARQL Endpoint. This complete path of scientific experiments is visualized as linked data with the help of two different views. The Project Dashboard provides a complete overview of all the experiments conducted for a project. Additionally, the ProvTrack module provides an interactive graph view of a complete path of an experiment. CAESAR is currently used by scientists in the ReceptorLight project in their daily research work for scientific data management. We evaluated the complete approach of provenance-based semantic approach by answering the competency questions using the data uploaded by scientists. The user-based evaluation also showed that CAESAR is useful for tracking the provenance of scientific experiments. Since most scientists do not possess the knowledge of SPARQL, such a complete view of the provenance of scientific experiments could not have been gained without the Dashboard and ProvTrack. This contribution helped to achieve Goal3 by validating the main hypothesis 2.3.

8.3 Future Work

In this thesis, we followed a provenance-based semantic approach for the understandability, reproducibility, and reuse of scientific experiments. We provided a provenance data model to describe the complete path of the scientific experiments by linking their different aspects. The ideas and concepts developed in this thesis are implemented in life-science experiments dealing with microscopy images. We expect that this approach can be extended to different types of experiments in diverse scientific disciplines. In addition to that, there is much room for extending our work. The modules developed were focused more on providing the feature. Improving the system for performance is one of the future lines of work. One part of the provenance capture module depends on the scientists to document their experimental data. Even though the metadata from the images capture the execution environment and the settings of the devices, the need for human annotations to the experimental datasets is extremely important. Besides this limitation, the mappings for the ontology-based data access required some manual curation. This can affect when the database is extended for other experiment types.

Additionally, we identify some future lines of work where this research can be extended in several ways:

- **Provenance data differencing of scientific experiments.**
  Currently, we provide a feature to compare and see the differences in the provenance of different executions of computational steps using ProvBook. We also provide basic provenance differencing of different versions of scientific experiment descriptions in CAESAR. This can be extended further to compare the complete path of multiple experiments. This could be implemented in ProvTrack where a user can choose experiments to visualize the comparison. The users could select two versions of an experiment or two different similar experiments. The users would be able to compare their experiments with other experiments from their team members in the collaborative environment. They could also compare different cases of reproducible experiments and see where the provenance graphs resulted in the divergence of results.

The PDiff algorithm can be applied in this case to see the divergence in the provenance graphs [Missier et al., 2016]. The provenance graph described using the REPRODUCE-ME ontology can be used to implement this algorithm and provide the visualization in ProvTrack.

- **Semantic Search**
  Currently, CAESAR provides an interface with a basic keyword search for the provenance information of the experiment. Even though querying the provenance information using SPARQL is possible through the SPARQL editor interface, most of the scientists do not possess SPARQL knowledge. So the search interface could be extended further to provide semantic search to look for similar kind of experimental datasets. In a multi-user environment, it would be helpful to see similar datasets and results which could lead to more collaboration. This could be further expanded to provide recommendations for scientists.

- **Reproducibility Checker Button**
  Based on the understanding of current research practices and experimental workflow followed by scientists, completely automated provenance capturing and management solution for scientific experiments is something which is difficult to achieve in the present state. Hence, we assume that reproducibility is not a one-button solution in the use cases that we provided in this thesis. It requires involvement and interactions of users especially through different non-computational steps of an experiment. Therefore, a reproducibility button to reproduce an experiment is currently not feasible unless every step is machine-controlled. A feasible solution is to have a reproducibility checker button for computational experiments which could provide whether the experiment could be reproduced using the current environment. It could provide intelligent decisions whether the list of all data needed for the experiment is present and if the result of the previous trials matched the result from the current trial.

- **Extending ProvBook**
  Currently, ProvBook provides provenance information of different executions of Jupyter Notebooks with several other features. Further work needs to be done in ProvBook to see how data and code inside each cell influence the results. In addition to that, a study needs to be conducted to analyze the effect of the execution order of the cells to the intermediate and final results.

- **Extending CAESAR**
  There are several possibilities to extend and improve CAESAR. The provenance management system could be expanded to include more types of experiments and experiment materials. Based on other ontologies, the metadata

editor could be extended to provide intelligent authoring and auto-completion of data. Another direction for future work is to use the existing provenance of results in CAESAR to design new experiments. In addition to that, several performance measures could be taken to reduce the query time for the SPARQL queries in the project dashboard and ProvTrack. The PAV ontology [Ciccarese et al., 2013], which also extends PROV could be used to tracking the provenance, authoring, and versioning of scientific experiments. The ProvTrack could be extended to visualize the evolution of experiments. CAESAR could be extended to serve as a public data repository providing DOIs to the experimental data along with the provenance information. This would help the scientific community to track the complete path of the provenance of the results described in the scientific publications.

Bibliography

[Ali and Moreau, 2013] Ali, M. and Moreau, L. (2013). A provenance-aware policy language (cprovl) and a data traceability model (cprov) for the cloud. In *2013 International Conference on Cloud and Green Computing*, pages 479–486. IEEE.

[Allan et al., 2012] Allan, C., Burel, J.-M., Moore, J., Blackburn, C., Linkert, M., Loynton, S., MacDonald, D., Moore, W. J., Neves, C., Patterson, A., et al. (2012). OMERO: flexible, model-driven data management for experimental biology. *Nature methods*, 9(3):245–253.

[Altintas et al., 2004] Altintas, I., Berkley, C., Jaeger, E., Jones, M. B., Ludäscher, B., and Mock, S. (2004). Kepler: An extensible system for design and execution of scientific workflows. In *Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM 2004), 21-23 June 2004, Santorini Island, Greece*, pages 423–424.

[Amstutz et al., 2016] Amstutz, P., Crusoe, M. R., Tijanić, N., Chapman, B., Chilton, J., Heuer, M., Kartashov, A., Leehr, D., Ménager, H., Nedeljkovich, M., et al. (2016). Common workflow language, v1. 0.

[Ashburner et al., 2000] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25.

[Atmanspacher and Maasen, 2016] Atmanspacher, H. and Maasen, S. (2016). *Reproducibility: principles, problems, practices, and prospects.* John Wiley & Sons.

[Baker, 2016] Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature News*, 533(7604):452.

[Bánáti et al., 2016] Bánáti, A., Kacsuk, P., and Kozlovszky, M. (2016). Investigation of the descriptors to make the scientific workflows reproducible. In *2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 000129–000134.

[Begley and Ioannidis, 2015] Begley, C. G. and Ioannidis, J. P. (2015). Reproducibility in science: improving the standard for basic and preclinical research. *Circulation research*, 116(1):116–126.

[Belhajjame et al., 2013] Belhajjame, K., B'Far, R., Cheney, J., Coppens, S., Cresswell, S., Gil, Y., Groth, P., Klyne, G., Lebo, T., McCusker, J., et al. (2013). PROV-DM: The PROV Data Model. *W3C Recommendation. http://www. w3. org/TR/prov-dm.*

[Belhajjame et al., 2015] Belhajjame, K., Zhao, J., Garijo, D., Gamble, M., Hettne, K., Palma, R., Mina, E., Corcho, O., Gmez-Prez, J. M., Bechhofer, S., Klyne, G., and Goble, C. (2015). Using a suite of ontologies for preserving workflow-centric research objects. *Web Semantics: Science, Services and Agents on the World Wide Web*, 32:16 – 42.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific american*, 284(5):28–37.

[Biskup et al., 2007] Biskup, C., Kusch, J., Schulz, E., Nache, V., Schwede, F., Lehmann, F., Hagen, V., and Benndorf, K. (2007). Relating ligand binding to activation gating in CNGA2 channels. *Nature*, 446(7134):440–443.

[Bizer et al., 2009] Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22.

[Boettiger, 2015] Boettiger, C. (2015). An introduction to docker for reproducible research. *SIGOPS Oper. Syst. Rev.*, 49(1):71–79.

[Borkin et al., 2013] Borkin, M. A., Yeh, C. S., Boyd, M., Macko, P., Gajos, K. Z., Seltzer, M. I., and Pfister, H. (2013). Evaluation of filesystem provenance visualization tools. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2476–2485.

[Brank et al., 2005] Brank, J., Grobelnik, M., and Mladenic, D. (2005). A survey of ontology evaluation techniques. In *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)*, pages 166–170. Citeseer Ljubljana, Slovenia.

[Brinkman et al., 2010] Brinkman, R. R., Courtot, M., Derom, D., Fostel, J., He, Y., Lord, P. W., Malone, J., Parkinson, H. E., Peters, B., Rocca-Serra, P., Ruttenberg, A., Sansone, S., Soldatova, L. N., Jr., C. J. S., Turner, J. A., Zheng, J., and et al. (2010). Modeling biomedical experimental processes with OBI. *J. Biomedical Semantics*, 1(S-1):S7.

[Brüggemann et al., 2016] Brüggemann, S., Bereta, K., Xiao, G., and Koubarakis, M. (2016). Ontology-based data access for maritime security. In Sack, H., Blomqvist, E., d'Aquin, M., Ghidini, C., Ponzetto, S. P., and Lange, C., editors, *The Semantic Web. Latest Advances and New Domains*, pages 741–757, Cham. Springer International Publishing.

[Callahan et al., 2006] Callahan, S. P., Freire, J., Santos, E., Scheidegger, C. E., Silva, C. T., and Vo, H. T. (2006). VisTrails: Visualization meets Data Management. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, SIGMOD '06, pages 745–747, New York, NY, USA. ACM.

[Calvanese et al., 2017] Calvanese, D., Cogrel, B., Komla-Ebri, S., Kontchakov, R., Lanti, D., Rezk, M., Rodriguez-Muro, M., and Xiao, G. (2017). Ontop: Answering SPARQL queries over relational databases. *Semantic Web*, 8(3):471–487.

[Cao et al., 2014] Cao, Y., Jones, C., Cuevas-Vicenttın, V., Jones, M. B., Ludäscher, B., McPhillips, T., Missier, P., Schwalm, C., Slaughter, P., Vieglais, D., et al. (2014). ProvONE: extending PROV to support the DataONE scientific community.

[Carvalho et al., 2016] Carvalho, L. A. M. C., Belhajjame, K., and Medeiros, C. B. (2016). Converting scripts into reproducible workflow research objects. In *2016 IEEE 12th International Conference on e-Science (e-Science)*, pages 71–80.

[Carvalho et al., 2017] Carvalho, L. A. M. C., Wang, R., Gil, Y., and Garijo, D. (2017). NiW: Converting notebooks into workflows to capture dataflow and provenance. SciKnow 2017, Austin, Texas, 2017.

[Chapman et al., 2008] Chapman, A. P., Jagadish, H. V., and Ramanan, P. (2008). Efficient provenance storage. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 993–1006, New York, NY, USA. ACM.

[Chirigati and Freire, 2017] Chirigati, F. and Freire, J. (2017). *Provenance and Reproducibility*, pages 1–5. Springer New York, New York, NY.

[Chirigati et al., 2013] Chirigati, F., Shasha, D., and Freire, J. (2013). ReproZip: Using provenance to support computational reproducibility. In *Presented as part of the 5th USENIX Workshop on the Theory and Practice of Provenance*, Lombard, IL. USENIX.

[Ciccarese et al., 2013] Ciccarese, P., Soiland-Reyes, S., Belhajjame, K., Gray, A. J., Goble, C., and Clark, T. (2013). PAV ontology: provenance, authoring and versioning. *Journal of Biomedical Semantics*, 4(1):37.

[Ciccarese et al., 2008] Ciccarese, P., Wu, E., Wong, G., Ocana, M., Kinoshita, J., Ruttenberg, A., and Clark, T. (2008). The SWAN biomedical discourse ontology. *Journal of Biomedical Informatics*, 41(5):739 – 751. Semantic Mashup of Biomedical Data.

[Cohen-Boulakia et al., 2017] Cohen-Boulakia, S., Belhajjame, K., Collin, O., Chopard, J., Froidevaux, C., Gaignard, A., Hinsen, K., Larmande, P., Bras, Y. L., Lemoine, F., Mareuil, F., Mnager, H., Pradal, C., and Blanchet, C. (2017). Scientific workflows for computational reproducibility in the life sciences: Status, challenges and opportunities. *Future Generation Computer Systems*, 75:284 – 298.

[Compton et al., 2012] Compton, M., Barnaghi, P., Bermudez, L., Garca-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W. D., Phuoc, D. L., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., and Taylor, K. (2012). The SSN ontology of the W3C semantic sensor network incubator group. *Journal of Web Semantics*, 17:25 – 32.

[Cyganiak et al., 2014] Cyganiak, R., Wood, D., Lanthaler, M., Klyne, G., Carroll, J. J., and McBride, B. (2014). RDF 1.1 concepts and abstract syntax. *W3C recommendation*, 25(02).

[Das et al., 2012] Das, S., Sundara, S., and Cyganiak, R. (2012). R2RML: RDB to RDF Mapping Language, W3C Recommendation, 27 september 2012. *Cambridge, MA: World Wide Web Consortium (W3C)(www. w3. org/TR/r2rml)*.

[Davidson et al., 2007] Davidson, S. B., Boulakia, S. C., Eyal, A., Ludäscher, B., McPhillips, T. M., Bowers, S., Anand, M. K., and Freire, J. (2007). Provenance in scientific workflow systems. *IEEE Data Eng. Bull.*, 30(4):44–50.

[Davidson and Freire, 2008] Davidson, S. B. and Freire, J. (2008). Provenance and scientific workflows: challenges and opportunities. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1345–1350. ACM.

[Davison, 2012] Davison, A. (2012). Automated capture of experiment context for easier reproducibility in computational research. *Computing in Science Engineering*, 14(4):48–56.

[Deelman et al., 2005] Deelman, E., Singh, G., Su, M., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Vahi, K., Berriman, G. B., Good, J., Laity, A. C., Jacob, J. C., and Katz, D. S. (2005). Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming*, 13(3):219–237.

[Degtyarenko et al., 2007] Degtyarenko, K., De Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcántara, R., Darsow, M., Guedj, M., and Ashburner, M. (2007). ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic acids research*, 36(suppl_1):D344–D350.

[Eliceiri et al., 2012] Eliceiri, K. W., Berthold, M. R., Goldberg, I. G., Ibáñez, L., Manjunath, B. S., Martone, M. E., Murphy, R. F., Peng, H., Plant, A. L., Roysam, B., et al. (2012). Biological imaging software tools. *Nature methods*, 9(7):697.

[Ferro and Silvello, 2017] Ferro, N. and Silvello, G. (2017). The road towards reproducibility in science: The case of data citation. In *Digital Libraries and Archives - 13th Italian Research Conference on Digital Libraries, IRCDL 2017, Modena, Italy, January 26-27, 2017, Revised Selected Papers*, pages 20–31.

[Fessakis et al., 2013] Fessakis, G., Gouli, E., and Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63:87–97.

[Freire et al., 2012] Freire, J., Bonnet, P., and Shasha, D. E. (2012). Computational reproducibility: state-of-the-art, challenges, and database research opportunities. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 593–596.

[Freire and Chirigati, 2018] Freire, J. and Chirigati, F. S. (2018). Provenance and the different flavors of reproducibility. *IEEE Data Eng. Bull.*, 41(1):15–26.

[Freire et al., 2016] Freire, J., Fuhr, N., and Rauber, A. (2016). Reproducibility of Data-Oriented Experiments in e-Science (Dagstuhl Seminar 16041). *Dagstuhl Reports*, 6(1):108–159.

[Freire et al., 2008] Freire, J., Koop, D., Santos, E., and Silva, C. T. (2008). Provenance for computational tasks: A survey. *Computing in Science and Engineering*, 10(3):11–21.

[Frew et al., 2008] Frew, J., Metzger, D., and Slaughter, P. (2008). Automatic capture and reconstruction of computational provenance. *Concurrency and Computation: Practice and Experience*, 20(5):485–496.

[Garijo, 2017] Garijo, D. (2017). WIDOCO: A wizard for documenting ontologies. In d'Amato, C., Fernandez, M., Tamma, V., Lecue, F., Cudré-Mauroux, P., Sequeda, J., Lange, C., and Heflin, J., editors, *The Semantic Web – ISWC 2017*, pages 94–102, Cham. Springer International Publishing.

[Garijo and Gil, 2011] Garijo, D. and Gil, Y. (2011). A new approach for publishing workflows: Abstractions, standards, and linked data. In *Proceedings of the 6th*

Workshop on Workflows in Support of Large-scale Science, WORKS ’11, pages 47–56, New York, NY, USA. ACM.

[Garijo and Gil, 2012] Garijo, D. and Gil, Y. (2012). Augmenting PROV with plans in P-Plan: scientific processes as linked data. CEUR Workshop Proceedings.

[Giraldo et al., 2014] Giraldo, O. L., Castro, A. G., and Corcho, Ó. (2014). SMART protocols: Semantic representation for experimental protocols. In *Proceedings of the 4th Workshop on Linked Science 2014 - Making Sense Out of Data (LISC2014) co-located with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Italy, October 19, 2014*, pages 36–47.

[Goble et al., 2010] Goble, C. A., Bhagat, J., Aleksejevs, S., Cruickshank, D., Michaelides, D. T., Newman, D. R., Borkum, M., Bechhofer, S., Roos, M., Li, P., and Roure, D. D. (2010). myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Research*, 38(Web-Server-Issue):677–682.

[Goddard and Melville, 2004] Goddard, W. and Melville, S. (2004). *Research methodology: An introduction*. Juta and Company Ltd.

[Goecks et al., 2010] Goecks, J., Nekrutenko, A., and Taylor, J. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology*, 11(8):R86.

[Gonçalves et al., 2017] Gonçalves, R. S., O’Connor, M. J., Romero, M. M., Egyedi, A. L., Willrett, D., Graybeal, J., and Musen, M. A. (2017). The CEDAR workbench: An ontology-assisted environment for authoring metadata that describe scientific experiments. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II*, pages 103–110.

[González-Beltrán et al., 2014] González-Beltrán, A., Maguire, E., Sansone, S.-A., and Rocca-Serra, P. (2014). linkedISA: semantic representation of ISA-Tab experimental metadata. *BMC Bioinformatics*, 15(14):S4.

[Gray et al., 2017] Gray, A. J. G., Goble, C. A., and Jimenez, R. (2017). Bioschemas: From potato salad to protein annotation. In *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017*.

[Groth and Moreau, 2013] Groth, P. and Moreau, L. (2013). Prov-Overview. an overview of the PROV family of documents. Project report.

[Groth et al., 2010] Groth, P. T., Gibson, A., and Velterop, J. (2010). The anatomy of a nanopublication. *Inf. Services and Use*, 30(1-2):51–56.

[Grüninger and Fox, 1995] Grüninger, M. and Fox, M. S. (1995). Methodology for the design and evaluation of ontologies. *Citeseer*.

[Guo and Seltzer, 2012] Guo, P. J. and Seltzer, M. (2012). BURRITO: wrapping your lab notebook in computational infrastructure. In *4th Workshop on the Theory and Practice of Provenance, TaPP'12, Boston, MA, USA, June 14-15, 2012*.

[Gupta, 2009] Gupta, A. (2009). *Data Provenance*, pages 608–608. Springer US, Boston, MA.

[Herschel et al., 2017] Herschel, M., Diestelkämper, R., and Ben Lahmar, H. (2017). A survey on provenance: What for? what form? what from? *The VLDB Journal*, 26(6):881–906.

[Hirschberg, 1977] Hirschberg, D. S. (1977). Algorithms for the longest common subsequence problem. *Journal of the ACM (JACM)*, 24(4):664–675.

[Hoekstra and Groth, 2015] Hoekstra, R. and Groth, P. (2015). PROV-O-Viz - Understanding the role of activities in provenance. In Ludäscher, B. and Plale, B., editors, *Provenance and Annotation of Data and Processes*, pages 215–220, Cham. Springer International Publishing.

[Holsapple and Joshi, 2002] Holsapple, C. W. and Joshi, K. D. (2002). A collaborative approach to ontology design. *Commun. ACM*, 45(2):42–47.

[Hutson, 2018] Hutson, M. (2018). Artificial intelligence faces reproducibility crisis. *Science*, 359(6377):725–726.

[Jupp et al., 2016] Jupp, S., Malone, J., Burdett, T., Heriche, J.-K., Williams, E., Ellenberg, J., Parkinson, H., and Rustici, G. (2016). The cellular microscopy phenotype ontology. *Journal of Biomedical Semantics*, 7(1):28.

[Kaiser, 2015] Kaiser, J. (2015). The cancer test. *Science*, 348(6242):1411–1413.

[Khan et al., 2019] Khan, F. Z., Soiland-Reyes, S., Sinnott, R. O., Lonie, A., Goble, C., and Crusoe, M. R. (2019). Sharing interoperable workflow provenance: A review of best practices and their practical application in CWLProv. *GigaScience*, 8(11). giz095.

[Kharlamov et al., 2017] Kharlamov, E., Hovland, D., Skjæveland, M. G., Bilidas, D., Jiménez-Ruiz, E., Xiao, G., Soylu, A., Lanti, D., Rezk, M., Zheleznyakov, D., Giese, M., Lie, H., Ioannidis, Y. E., Kotidis, Y., Koubarakis, M., and Waaler, A. (2017). Ontology based data access in Statoil. *J. Web Semant.*, 44:3–36.

[Kluyver et al., 2016] Kluyver, T., Ragan-Kelley, B., et al. (2016). Jupyter notebooks-a publishing format for reproducible computational workflows. In *ELPUB*, pages 87–90.

[Knuth, 1984] Knuth, D. E. (1984). Literate programming. *Comput. J.*, 27(2):97–111.

[Kobayashi et al., 2018] Kobayashi, N., Kume, S., Lenz, K., and Masuya, H. (2018). RIKEN metadatabase: A database platform for health care and life sciences as a microcosm of linked open data cloud. *Int. J. Semantic Web Inf. Syst.*, 14(1):140–164.

[Kume et al., 2016] Kume, S., Masuya, H., Kataoka, Y., and Kobayashi, N. (2016). Development of an Ontology for an Integrated Image Analysis Platform to enable Global Sharing of Microscopy Imaging Data. In *Proceedings of the ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference.*

[Kunde et al., 2008] Kunde, M., Bergmeyer, H., and Schreiber, A. (2008). Requirements for a provenance visualization component. In Freire, J., Koop, D., and Moreau, L., editors, *Provenance and Annotation of Data and Processes*, pages 241–252, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Küster et al., 2011] Küster, M. W., Ludwig, C., Al-Hajj, Y., and Selig, T. (2011). TextGrid provenance tools for digital humanities ecosystems. In *5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011)*, pages 317–323.

[Kvilekval et al., 2010] Kvilekval, K., Fedorov, D., Obara, B., Singh, A., and Manjunath, B. (2010). Bisque: a platform for bioimage analysis and management. *Bioinformatics*, 26(4):544–552.

[Lanthaler and Gütl, 2012] Lanthaler, M. and Gütl, C. (2012). On using JSON-LD to create evolvable restful services. In *Third International Workshop on RESTful Design, WS-REST '12, Lyon, France, April 16, 2012*, pages 25–32.

[Lebo et al., 2013] Lebo, T., Sahoo, S., McGuinness, D., Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S., Zednik, S., and Zhao, J. (2013). PROV-O: The PROV Ontology. *W3C Recommendation*, 30.

[Lee et al., 2006] Lee, B., Plaisant, C., Parr, C. S., Fekete, J.-D., and Henry, N. (2006). Task taxonomy for graph visualization. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV '06, pages 1–5, New York, NY, USA. ACM.

[Linkert et al., 2010] Linkert, M., Rueden, C. T., Allan, C., Burel, J.-M., Moore, W., Patterson, A., Loranger, B., Moore, J., Neves, C., MacDonald, D., et al. (2010). Metadata matters: access to image data in the real world. *The Journal of cell biology*, 189(5):777–782.

[Liu et al., 2015] Liu, J., Pacitti, E., Valduriez, P., and Mattoso, M. (2015). A survey of data-intensive scientific workflow management. *J. Grid Comput.*, 13(4):457–493.

[Macko and Seltzer, 2011] Macko, P. and Seltzer, M. (2011). Provenance Map Orbiter: Interactive exploration of large provenance graphs. In *3rd Workshop on the Theory and Practice of Provenance, TaPP'11, Heraklion, Crete, Greece, June 20-21, 2011*.

[Malone et al., 2014] Malone, J., Brown, A., Lister, A. L., Ison, J., Hull, D., Parkinson, H., and Stevens, R. (2014). The Software Ontology (SWO): a resource for reproducibility in biomedical data analysis, curation and digital preservation. *Journal of Biomedical Semantics*, 5(1):25.

[Mayer et al., 2012] Mayer, R., Rauber, A., Neumann, M. A., Thomson, J., and Antunes, G. (2012). Preserving scientific processes from design to publications. In Zaphiris, P., Buchanan, G., Rasmussen, E., and Loizides, F., editors, *Theory and Practice of Digital Libraries*, pages 113–124, Berlin, Heidelberg. Springer Berlin Heidelberg.

[McGuinness et al., 2004] McGuinness, D. L., Van Harmelen, F., et al. (2004). OWL web ontology language overview. *W3C recommendation*, 10(10):2004.

[McPhillips et al., 2015] McPhillips, T., Song, T., Kolisnik, T., Aulenbach, S., Belhajjame, K., Bocinsky, K., Cao, Y., Chirigati, F., Dey, S., Freire, J., et al. (2015). YesWorkflow: a user-oriented, language-independent tool for recovering workflow information from scripts. *arXiv preprint arXiv:1502.02403*.

[Meester et al., 2016] Meester, D., Dimou, Verborgh, Mannens, and Walle (2016). An ontology to semantically declare and describe functions. In Sack, H., Rizzo, G., Steinmetx, N., Mladenić, D., Auer, S., and Lange, C., editors, *The Semantic Web; ESWC 2016 Satellite Events*, volume 9989 of *Lecture Notes in Computer Science*, pages 46–49. Springer International Publishing.

[Miao and Deshpande, 2018] Miao, H. and Deshpande, A. (2018). ProvDB: Provenance-enabled lifecycle management of collaborative data analysis workflows. *IEEE Data Eng. Bull.*, 41(4):26–38.

[Michener et al., 2011] Michener, W., Vieglais, D., Vision, T., Kunze, J., Cruse, P., and Janée, G. (2011). DataONE: Data Observation Network for EarthPreserving data and enabling innovation in the biological and environmental sciences. *D-Lib Magazine*, 17(1/2):12.

[Missier, 2016] Missier, P. (2016). The lifecycle of provenance metadata and its associated challenges and opportunities. In *Building Trust in Information*, pages 127–137. Springer.

[Missier et al., 2013] Missier, P., Dey, S. C., Belhajjame, K., Cuevas-Vicenttín, V., and Ludäscher, B. (2013). D-PROV: extending the PROV provenance model with workflow structure. In *5th Workshop on the Theory and Practice of Provenance, TaPP'13, Lombard, IL, USA, April 2-3, 2013*.

[Missier et al., 2016] Missier, P., Woodman, S., Hiden, H., and Watson, P. (2016). Provenance and data differencing for workflow reproducibility analysis. *Concurrency and Computation: Practice and Experience*, 28(4):995–1015.

[Moreau, 2010] Moreau, L. (2010). The foundations for provenance on the web. *Foundations and Trends in Web Science*, 2(2–3):99–241.

[Moreau, 2011] Moreau, L. (2011). Provenance-based reproducibility in the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(2):202–221.

[Moreau et al., 2011] Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., and den Bussche, J. V. (2011). The Open Provenance Model core specification (v1.1). *Future Generation Computer Systems*, 27(6):743 – 756.

[Moreau et al., 2015] Moreau, L., Groth, P. T., Cheney, J., Lebo, T., and Miles, S. (2015). The rationale of PROV. *J. Web Semant.*, 35:235–257.

[Moreau et al., 2008] Moreau, L., Ludäscher, B., Altintas, I., Barga, R. S., Bowers, S., Callahan, S. P., Jr., G. C., Clifford, B., Cohen, S., Boulakia, S. C., Davidson, S. B., Deelman, E., Digiampietri, L. A., Foster, I. T., Freire, J., Frew, J., Futrelle, J., Gibson, T., Gil, Y., Goble, C. A., Golbeck, J., Groth, P. T., Holland, D. A., Jiang, S., Kim, J., Koop, D., Krenek, A., McPhillips, T. M., Mehta, G., Miles, S., Metzger, D., Munroe, S., Myers, J., Plale, B., Podhorszki, N., Ratnakar, V., Santos, E., Scheidegger, C. E., Schuchardt, K., Seltzer, M. I., Simmhan, Y. L., Silva, C. T., Slaughter, P., Stephan, E. G., Stevens, R., Turi, D., Vo, H. T., Wilde, M., Zhao, J., and Zhao, Y. (2008). Special issue: The first provenance challenge. *Concurrency and Computation: Practice and Experience*, 20(5):409–418.

[Moreau and Tranchevent, 2012] Moreau, Y. and Tranchevent, L.-C. (2012). Computational tools for prioritizing candidate genes: boosting disease gene discovery. *Nature Reviews Genetics*, 13(8):523.

[Muniswamy-Reddy et al., 2006] Muniswamy-Reddy, K., Holland, D. A., Braun, U., and Seltzer, M. I. (2006). Provenance-aware storage systems. In *Proceedings of the 2006 USENIX Annual Technical Conference, Boston, MA, USA, May 30 - June 3, 2006*, pages 43–56.

[Murta et al., 2014] Murta, L., Braganholo, V., Chirigati, F., Koop, D., and Freire, J. (2014). noWorkflow: capturing and analyzing provenance of scripts. In *International Provenance and Annotation Workshop*, pages 71–83. Springer.

[Myers, 1986] Myers, E. W. (1986). An O(ND) difference algorithm and its variations. *Algorithmica*, 1(1-4):251–266.

[Niles and Pease, 2001] Niles, I. and Pease, A. (2001). Towards a Standard Upper Ontology. In *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, pages 2–9, New York, NY, USA. ACM.

[Noy et al., 2001] Noy, N. F., McGuinness, D. L., et al. (2001). Ontology development 101: A guide to creating your first ontology. Technical report.

[Oinn et al., 2004] Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M. R., Wipat, A., et al. (2004). Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Peng, 2015] Peng, R. (2015). The reproducibility crisis in science: A statistical counterattack. *Significance*, 12(3):30–32.

[Pérez and Pérez-Hernández, 2015] Pérez, I. S. and Pérez-Hernández, M. S. (2015). Towards reproducibility in scientific workflows: An infrastructure-based approach. *Scientific Programming*, 2015:243180:1–243180:11.

[Pimentel et al., 2017] Pimentel, J. a. F., Murta, L., Braganholo, V., and Freire, J. (2017). noWorkflow: A tool for collecting, analyzing, and managing provenance from Python scripts. *Proc. VLDB Endow.*, 10(12):1841–1844.

[Pimentel et al., 2019] Pimentel, J. a. F., Murta, L., Braganholo, V., and Freire, J. (2019). A large-scale study about quality and reproducibility of jupyter notebooks. In *Proceedings of the 16th International Conference on Mining Software Repositories*, MSR '19, pages 507–517, Piscataway, NJ, USA. IEEE Press.

[Pimentel et al., 2015] Pimentel, J. F. N., Braganholo, V., Murta, L., and Freire, J. (2015). Collecting and analyzing provenance on interactive notebooks: When IPython meets noWorkflow. In *7th USENIX Workshop on the Theory and Practice of Provenance (TaPP 15)*, Edinburgh, Scotland. USENIX Association.

[Poggi et al., 2008] Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., and Rosati, R. (2008). Linking data to ontologies. *J. Data Semantics*, 10:133–173.

[Prud'hommeaux and Seaborne, 2008] Prud'hommeaux, E. and Seaborne, A. (2008). SPARQL Query Language for RDF. W3C Recommendation. `http://www.w3.org/TR/rdf-sparql-query/`.

[Ram and Liu, 2006] Ram, S. and Liu, J. (2006). Understanding the semantics of data provenance to support active conceptual modeling. In *Active Conceptual Modeling of Learning, Next Generation Learning-Base System Development [1st International ACM-L Workshop, November 8, 2006, during ER 2006, Tucson, Arizona, USA].*, pages 17–29.

[Rodriguez-Muro et al., 2013] Rodriguez-Muro, M., Kontchakov, R., and Zakharyaschev, M. (2013). Ontology-based data access: Ontop of databases. In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, pages 558–573.

[Rule et al., 2018] Rule, A., Tabard, A., and Hollan, J. D. (2018). Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 32:1–32:12, New York, NY, USA. ACM.

[Sahoo, 2010] Sahoo, S. S. (2010). Semantic provenance: Modeling, querying, and application in scientific discovery.

[Sahoo et al., 2008] Sahoo, S. S., Sheth, A. P., and Henson, C. A. (2008). Semantic provenance for eScience: Managing the deluge of scientific data. *IEEE Internet Computing*, 12(4):46–54.

[Sahoo et al., 2019] Sahoo, S. S., Valdez, J., Kim, M., Rueschman, M., and Redline, S. (2019). ProvCaRe: Characterizing scientific reproducibility of biomedical research studies using semantic provenance metadata. *I. J. Medical Informatics*, 121:10–18.

[Samuel, 2017] Samuel, S. (2017). Integrative data management for reproducibility of microscopy experiments. In *The Semantic Web - 14th International Conference,* *ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part II*, pages 246–255.

[Samuel, 2019] Samuel, S. (2019). REPRODUCE-ME. `https://w3id.org/reproduceme/research`.

[Samuel et al., 2018] Samuel, S., Groeneveld, K., Taubert, F., Walther, D., Kache, T., Langenstück, T., König-Ries, B., Bücker, H. M., and Biskup, C. (2018). The Story of an experiment: a provenance-based semantic approach towards research reproducibility. In *Proceedings of the 11th International Conference Semantic Web Applications and Tools for Life Sciences, SWAT4LS 2018, Antwerp, Belgium, December 3-6, 2018.*

[Samuel and König-Ries, 2017] Samuel, S. and König-Ries, B. (2017). REPRODUCE-ME: ontology-based data access for reproducibility of microscopy experiments. In *The Semantic Web: ESWC 2017 Satellite Events - ESWC 2017 Satellite Events, Portorož, Slovenia, May 28 - June 1, 2017, Revised Selected Papers*, pages 17–20.

[Samuel and König-Ries, 2018a] Samuel, S. and König-Ries, B. (2018a). Combining P-Plan and the REPRODUCE-ME ontology to achieve semantic enrichment of scientific experiments using interactive notebooks. In *The Semantic Web: ESWC 2018 Satellite Events - ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers*, pages 126–130.

[Samuel and König-Ries, 2018b] Samuel, S. and König-Ries, B. (2018b). ProvBook: Provenance-based semantic enrichment of interactive notebooks for reproducibility. In *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, USA, October 8th to 12th, 2018.*

[Samuel and König-Ries, 2018c] Samuel, S. and König-Ries, B. (2018c). ProvBook: Provenance of the Notebook. `https://doi.org/10.6084/m9.figshare.6401096.v1`.

[Samuel and König-Ries, 2019] Samuel, S. and König-Ries, B. (2019). Survey on Understanding Experiments and Research Practices for Reproducibility: Material and Results. `https://doi.org/10.6084/m9.figshare.8313782.v1`.

[Samuel et al., 2017] Samuel, S., Taubert, F., Walther, D., König-Ries, B., and Bücker, H. M. (2017). Towards reproducibility of microscopy experiments. *D-Lib Magazine*, 23(1/2).

[Sandve et al., 2013] Sandve, G. K., Nekrutenko, A., Taylor, J., and Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLOS Computational Biology*, 9(10):1–4.

[Sarikhani and Wendelborn, 2018] Sarikhani, M. and Wendelborn, A. L. (2018). Mechanisms for provenance collection in scientific workflow systems. *Computing*, 100(5):439–472.

[Scheidegger et al., 2008] Scheidegger, C. E., Vo, H. T., Koop, D., Freire, J., and Silva, C. T. (2008). Querying and re-using workflows with VsTrails. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1251–1254. ACM.

[Shen, 2014] Shen, H. (2014). Interactive notebooks: Sharing the code. *Nature News*, 515(7525):151.

[Simmhan et al., 2011] Simmhan, Y., Groth, P. T., and Moreau, L. (2011). Special section: The third provenance challenge on using the open provenance model for interoperability. *Future Generation Comp. Syst.*, 27(6):737–742.

[Simmhan et al., 2005] Simmhan, Y., Plale, B., and Gannon, D. (2005). A survey of data provenance in e-science. *SIGMOD Record*, 34(3):31–36.

[Soldatova and King, 2006] Soldatova, L. and King, R. (2006). An ontology of scientific experiments. *Journal of the Royal Society Interface*, 3(11):795–803.

[Spjuth et al., 2015] Spjuth, O., Bongcam-Rudloff, E., Hernández, G. C., Forer, L., Giovacchini, M., Guimera, R. V., Kallio, A., Korpelainen, E., Kańduła, M. M., Krachunov, M., Kreil, D. P., Kulev, O., Łabaj, P. P., Lampa, S., Pireddu, L., Schönherr, S., Siretskiy, A., and Vassilev, D. (2015). Experiences with workflows for automating data-intensive bioinformatics. *Biology Direct*, 10(1):43.

[Steele and Iliinsky, 2011] Steele, J. and Iliinsky, N. (2011). *Designing Data Visualizations*. O'Reilly Media, Inc.

[Stitz et al., 2016] Stitz, H., Luger, S., Streit, M., and Gehlenborg, N. (2016). Avocado: Visualization of workflow-derived data provenance for reproducible biomedical research. *Comput. Graph. Forum*, 35(3):481–490.

[Studer et al., 1998] Studer, R., Benjamins, V., and Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1):161 – 197.

[Suárez-Figueroa et al., 2009] Suárez-Figueroa, M. C., Gómez-Pérez, A., and Villazón-Terrazas, B. (2009). How to write and use the ontology requirements specification document. In Meersman, R., Dillon, T., and Herrero, P., editors, *On* *the Move to Meaningful Internet Systems: OTM 2009*, pages 966–982, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Tariq et al., 2012] Tariq, D., Ali, M., and Gehani, A. (2012). Towards automated collection of application-level data provenance. In *4th Workshop on the Theory and Practice of Provenance, TaPP'12, Boston, MA, USA, June 14-15, 2012.*

[Taylor and Kuyatt, 1994] Taylor, B. N. and Kuyatt, C. E. (1994). Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results. Technical report, NIST Technical Note 1297.

[Taylor et al., 2010] Taylor, C., Field, D., Maguire, E., Begley, K., Brandizi, M., Sklyar, N., Hofmann, O., Sterk, P., Rocca-Serra, P., Neumann, S., Harris, S., Sansone, S.-A., Tong, W., and Hide, W. (2010). ISA software suite: supporting standards-compliant experimental annotation and enabling curation at the community level. *Bioinformatics*, 26(18):2354–2356.

[Taylor et al., 2008] Taylor, C. F., Field, D., Sansone, S.-A., Aerts, J., Apweiler, R., Ashburner, M., Ball, C. A., Binz, P.-A., Bogue, M., Booth, T., et al. (2008). Promoting coherent minimum reporting guidelines for biological and biomedical investigations: The MIBBI project. *Nature biotechnology*, 26(8):889.

[Team et al., 2015] Team, R. et al. (2015). RStudio: integrated development for R. *RStudio, Inc., Boston, MA URL http://www. rstudio. com*, 42.

[Wilkinson et al., 2016] Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data*, 3.

[Williams et al., 2017] Williams, E., Moore, J., Li, S. W., Rustici, G., Tarkowska, A., Chessel, A., Leo, S., Antal, B., Ferguson, R. K., Sarkans, U., et al. (2017). Image Data Resource: a bioimage data integration and publication platform. *Nature methods*, 14(8):775.

[Wolfram, 1988] Wolfram, S. (1988). *Mathematica: A System for Doing Mathematics by Computer.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Zhao et al., 2012] Zhao, J., Gómez-Pérez, J. M., Belhajjame, K., Klyne, G., García-Cuesta, E., Garrido, A., Hettne, K. M., Roos, M., Roure, D. D., and Goble, C. A. (2012). Why workflows break - understanding and combating decay in Taverna workflows. In *8th IEEE International Conference on E-Science, e-Science 2012, Chicago, IL, USA, October 8-12, 2012*, pages 1–9.

Appendices

Appendix A

Interviews with Scientists

These are the responses from the interviews with the scientists who attended the workshop on "Fostering reproducible science – What data management tools can do and should do for you" was conducted in conjunction with BEXIS2 UserDevConf[1] Conference.

> *My work involves the usage of sampling protocols and field work where data is collected (on paper). The collected data is transferred to Excel and are corrected for typos, errors, etc. It is then documented with metadata on a readme file. External hard drives are used for storing experimental data. Analysis of data is done using scripts. After the publication of the paper, the associated data are uploaded to public repositories. The data is not made available until the paper is not published. During this stage, it is difficult to get help from and collaborate with other researchers and colleagues distributed geographically. To facilitate reuse, well-documented structured data sets which are quality controlled is extremely required.* -P1, Ecologist

> *Version-controlled repositories are used to document my code. The code and the data used for experiments need to be version-controlled and should also be citable. The publication should clearly provide the pointer to the data and code that is used in the study. Space constraint is a problem with repositories for the gigabytes of data generated from the experiments.* -P2, Computer Scientist

> *Traditional lab books are used to document experiments. The methods used in the experiments are based on the author's first publication. Further information is gathered with direct communication via e-mail with other scientists. The experimental metadata is stored in external hard disks for the long-term preservation of data. Multiple copies are*

[1] `http://fusion.cs.uni-jena.de/bexis2userdevconf2017/workshop/`

stored in these drives. The information related to experiments is stored in different places/computers. Different software and devices are used in the experimental workflow. The version and license of the software are important to document in the experimental metadata. Since the experimental metadata are stored at different places including lab books, it gets hard to distinguish good experiments or to find a specific experiment. All the information at one place is much needed for the reproducibility and reusability of results. Another requirement is to crosslink the experiment metadata with the data and results. It should be possible to follow the crosslink to understand an experiment. In some cases, it is important to document the individual trials which will help to understand which possibilities did not work out. -P3, Biologist

Code and data which are required to reproduce results need to be publicly available. The ability to visualize and compare results helps to understand our own and others work. -P4, Computer Scientist

There are currently two problems faced by scientists. First, there is a lack of awareness of data management process. Second, even if there is data management awareness, there is a lack of resources for the proper management of data. Different data management methods and tools are followed in different teams and institutes. Lack of documenting the data in a structured manner causes problems in understanding experiments when people change teams. Proper guidelines for data management for big projects need to be maintained. -P5, Ecologist

It is important to share data within consortiums. Hence, distributed and centralized data management within a consortium is required. The volume of data generated by instruments is a concern, which needs to be addressed using data sharing policy. The documentation of procedures, devices, and steps are vital for reproducibility. -P6, Biochemist

My work involves data collection from the field, designing experiments and analyzing the data. R Scripts are used for the analysis of collected data. The data is stored in external hard disks. Manual documentation of the steps is done and stored in a text file. The results are stored as Excel tables. It is important to have the accessibility of non-published data among the colleagues in a group. It is also important to have simple tools for data management rather than complicated tools which require extra learning. -P7, Ecologist

Raw data and processed data are needed to understand an experiment. R Scripts are used for processing and simulation of statistical models.

Documenting what the scripts are doing is necessary to understand the results from the scripts. -P8, Mathematician

Data collection is an important part of my work which includes images of a geographical location. It is important to know the relationship between the experimental data like images to the geographical sites where the photo was taken. Thus a system providing a link to the data, metadata, and results can help in our daily work. -P9, Geologist

My work is to develop materials for chemical compounds which involves measurements with spectroscopy techniques and relies on statistical techniques. Data analysis using R and Matlab scripts is also an important part of my work. The documentation of experiments is done in normal lab notebooks. The need for documenting the experimental metadata is the first and foremost thing. In these traditional notebooks, it is very important to write everything clearly for other new scientists in the group to reproduce the experiment. It is also important to categorize the data including the processed data, analyzed data, raw data, external files, etc. A common data management repository is very important so to categorize and store this data so that even if a scientist leaves a group, the data is stored for future use. The comparison of data-to-data is also important. Documenting every step is essential for reproducibility. The methods in the publication are not sufficient to fully reproduce an experiment. -P10, Physicist

Appendix B

The REPRODUCE-ME Ontology Requirements Specification Document

The REPRODUCE-ME Ontology Requirements Specification Document
1. Purpose
The purpose of this ontology is to represent the provenance of a scientific experiment to enable end-to-end reproducibility.
2. Scope
The ontology has to focus on the computational and non-computational processes of an experiment and the data used and generated in an experiment.
3. Implementation Language
The ontology will be implemented in OWL language.
4. Intended End-Users
User 1. Scientist aiming to track the provenance of scientific experiments.
User 2. Scientist aiming for end-to-end reproducibility of scientific experiments.
User 3. Scientist aiming to track the provenance of execution of scripts.
User 4. Scientist aiming to track the provenance of execution of Computational notebooks.
User 5. Scientist aiming to describe light microscopy imaging experiments.
5. Intended Uses
Use 1. Describe the provenance of scientific experiments.
Use 2. Describe the computational experiments conducted using scripts.
Use 3. Describe the computational experiments conducted using interactive notebooks.
Use 4. Describe the steps and the execution environment of experiments.

Table B.1: The REPRODUCE-ME ORSD Slots 1-5

6. Ontology Requirements
a. Non-functional Requirements
NFR 1. The ontology must be published on the Web with an open and non-commercial license.
NFR 2. The ontology must be written in English.
NFR 3. The ontology must follow the Camel Case convention.
NFR 4. The ontology must be available via its namespace URI with human-readable documentation and machine-readable structured data using content negotiation. NFR 5. The ontology must reuse other ontologies if required.
b. Functional Requirements: Groups of Competency Questions
CQ1. What are the input and output variables of an experiment? **CQ2**. Which are the methods and standard operating procedures used? **CQ3**. Which are the files and materials that were used in a particular step? **CQ4**. Which are the steps involved in an experiment which used a particular material? **CQ5**. Which are the instruments that are associated with an experiment and their settings when the output was generated? **CQ6**. Which are the agents directly or indirectly responsible for an experiment? **CQ7**. Who created this experiment and when? Who modified it and when? **CQ8**. Which are the publications or external resources that were referenced? **CQ9**. What is the complete path taken by a scientist for an experiment? **CQ11**. What is the complete path taken by a user for a computational notebook experiment? **CQ12**. What is the sequence of steps in the execution of a computational notebook? **CQ13**. How many trials were performed for a particular cell in a computational notebook? **CQ14**. How long it took for a particular trial of a computational notebook? **CQ15**. What was the source for a particular trial of a computational notebook? **CQ16**. What was the output for a particular trial of a computational notebook? **CQ17**. Who are the agents responsible for the execution of a computational notebook? **CQ18**. When was a particular trial of a computational notebook last executed? **CQ19**. What are the environmental attributes of a notebook execution? **CQ20**. What is the sequence of steps in the execution of a script? **CQ21**. Which are the steps that invoke a particular module? **CQ22**. Which are the environmental attributes in the execution of a script? **CQ22**. List the user, the operating system, the processor, programming language version, the working directory associated with the execution of a script. **CQ24**. What is the complete derivation of a script output?

Table B.2: The REPRODUCE-ME ORSD Ontology Requirements

7. Pre-Glossary of Terms					
a. Terms from Competency Questions + Frequency					
Experiment	7	Step	6	Computation	8
Output	5	Input	2	Particular	8
Script	4	Notebook	8	Result	1
Setting	1	Complete	3	Execution	6
Trial	5	Sequence	2	Material	2
Agent	2	Environment	2	Instrument	1
File	1	Resource	1	Path	2
Attribute	2	Use	3	Responsible	2
Publication	1	Parameter	1	Procedure	1
Method	1	Version	1	Generate	2
b. Terms from Answers + Frequency					
Experiment	15	File	10	Code	4
Image	8	Plasmid	2	Protein	3
Microscope	4	Vector	3	Person	7
Setting	8	Solution	4	Data	6
Execution	6	Software	4	Material	6
Metadata	7	Format	3	Measurement	5
Time	2	Group	3	Instrument	3
Environment	2	Sample	2	Hardware	2
Result	7	Temperature	2	Project	1
Publication	4	Document	4	Cell	1
Version	3	Processed	2	Raw	1
c. Objects					
No objects were identified.					

Table B.3: The REPRODUCE-ME ORSD Pre-Glossary of Terms

Appendix C

Understanding Experiments and Research Practices for Reproducibility

Welcome to this survey.

The purpose of this study is to gain a better understanding of what is needed to achieve reproducibility of experiments in science. The results of this study will help us in developing tools that support reproducibility. In turn, this will (hopefully) benefit the scientific community.

This survey should take around 10 minutes to complete. This survey is completely anonymous. We provide this survey in the context of DFG CRC/TRR ReceptorLight.

If you have any questions regarding this survey, please contact Sheeba Samuel (sheeba.samuel@uni-jena.de) or Prof. Dr. Birgitta König-Ries (birgitta.koenig-ries@uni-jena.de).

Section A: Privacy Policy/Datenschutzerklärung

Dear scholar,

From 25th May 2018, the new General Data Protection Regulation (GDPR) (in German: Datenschutz-Grundverordnung, DSGVO) has come into effect. For compliance reasons, we are obliged to get your consent on the privacy policy before collecting any kind of personal information.

The following information applies as a supplement to the general privacy policy of the Friedrich Schiller University Jena. We request you to please kindly read both policies carefully and to agree.

Please note, as the Friedrich Schiller University Jena is headquartered in Germany, only the German version of this privacy policy is legally binding.

Liebe Wissenschaftlerinnen und Wissenschaftler,

Am 25. Mai 2018 trat die Datenschutz-Grundverordnung (DSGVO, https://dsgvo-gesetz.de/) in Kraft. Um Ihre personenbezogenen Daten erheben und verarbeiten zu können, benötigen wir daher aus rechtlichen Gründen zunächst Ihre Zustimmung zur Datenverarbeitung.

Die folgende Erklärung dient als Ergänzung zur Datenschutzerklärung der Friedrich-Schiller-Universität Jena. Wir möchten Sie bitten, beide Dokumente gründlich durchzulesen und mit Ihrer Einwilligung zu bestätigen, dass Sie der Datenverarbeitung zustimmen.

Privacy Policy (English) Datenschutzerklärung (Deutsch) Definition We use common language instead of more formal terms throughout this policy. To help ensure your understanding of some particular key terms, here is a table of translations:

When we say…

…we mean

“Friedrich Schiller University Jena”/”we”/”us”/”our” The Friedrich Schiller University Jena that conducts this survey. “this survey” The forms on this website that collect your answers. “personal information” Information you provide us or information we collect from you that could be used to personally identify you. We consider at least the following to be “personal information”:

IP address, operating system, browser

“third party” Individuals, entities, websites, services, products, and applications that are not controlled, managed, or operated by the Friederich Schiller University Jena Collection and Use of Information How do we collect personal information?

In this survey, we only collect data about your research contexts such as primary research fields and your research practice. All questions are not mandatory. You can omit questions you can not answer or you do not want to answer.

While browsing, some general information is stored in the server log files. We collect (1) the browser type and version used, (2) the operating system used by the accessing system (3) the date and time of access to the Internet site (4) and the Internet protocol address (IP address).

All this information is needed to deliver correct website content and to optimize web content continuously. In the case of cyber-attacks, log files provide necessary information for criminal prosecution.

How do we use that information?

We use your answers to gain better understanding of what is needed to achieve reproducibility of experiments in science and to understand the research practices followed in different science domain.

The results of this study will help us in developing tools, methods and workflows that support reproducibility. In turn, this will benefit the scientific community.

Sharing We support the idea of generating only FAIR data. Thus, we intend to publish all answers as open data in a data

Section B:

Research Context

In this section, we would like to know about your research background.

B1. **What is your current position?**

Student □

PhD Student □

Research Associate □

PostDoc □

Junior Research Group Leader/ Junior Professor □

Technical Assistant □

Lecturer □

Data Manager □

Professor □

Other □

Other

B2. What is your primary area of study?

Molecular Biology □

Cell Biology □

Microbiology □

Neuroscience □

Biology(other) □

Chemistry □

Plant Sciences □

Health Sciences □

Environmental Sciences □

Physics □

Computer Science □

Other □

Other

□

Section C:

Reproducibility

Reproducibility is the ability of getting the same (or close-by) results when repeating an experiment under different conditions of measurement (e.g. experimental setup, experimenter).

Reproducibility crisis refers to the growing belief that the results of many scientific studies are difficult or impossible to reproduce on subsequent investigation, either by independent researchers or by the original researchers themselves.

C1. Do you think there is a reproducibility crisis in your field of research?

Yes □

No □

Other □

Other

□

C2. In your experience, what are the factors leading to poor reproducibility?

Lack of sufficient metadata regarding the experiment (e.g. culturing conditions, environmental conditions, software version) □

Lack of data that is publicly available for use (e.g. code, methods, results) □

Lack of complete information in the Methods/Standard Operating Procedures/Protocols □

Poor experimental design □

Lack of resources like equipments/devices in your workplace □

Lack of the information related to the settings used in original experiment (eg. Experiment Setup, Instrument Settings) □

Difficulty in understanding laboratory notebook records □

Pressure to publish □

Lack of knowledge or training on reproducible research practices □

Lack of time to follow reproducible research practices □

Data privacy (e.g. Data sharing with third parties) □

Other □

Other

Section D:

Measures to ensure reproducibility

In this section, we would like to know about the measures taken in your field of research to ensure reproducibility.

D1. How easy would it be for you to find all the experimental data related to your own project in order to reproduce the results at a later point in time (e.g. 6 months after the original experiment)?

	Very Easy	Easy	Neither Easy nor difficult	Difficult	Very difficult
Input Data	□	□	□	□	□
Metadata about the methods	□	□	□	□	□
Metadata about the steps	□	□	□	□	□
Metadata about the experimental setup	□	□	□	□	□
Results	□	□	□	□	□

D2. How easy would it be for a newcomer in your workplace to find all the experimental data related to your project/experiment without any/limited instructions from you?

	Very Easy	Easy	Neither Easy nor difficult	Difficult	Very difficult
Input Data	□	□	□	□	□
Metadata about the methods	□	□	□	□	□
Metadata about the steps	□	□	□	□	□
Metadata about the experimental setup	□	□	□	□	□
Results	□	□	□	□	□

D3. Have you ever been unable to reproduce published results of others?

Yes □

No □

Never tried to reproduce others published results □

D4. Has anybody contacted you that they have a problem in reproducing your published results?

Yes □

No □

D5. Do you repeat your experiments to verify the results?

Yes □

No □

Sometimes □

Section E:

In order to reproduce published experiment results...

In this section, we would like to know the factors that are important for you to understand a scientific experiment in your field of research to enable reproducibility.

In order to reproduce published experiment results, what is your opinion on sharing metadata...?

E1. What is your opinion on sharing experimental data?

	Not Important At All	Little Importance	Average Importance	Very Important	Absolutely Essential	Not applicable
Raw Data	□	□	□	□	□	□
Processed Data	□	□	□	□	□	□
Negative Results	□	□	□	□	□	□
Measurements	□	□	□	□	□	□
Scripts/Code/Program	□	□	□	□	□	□
Image Annotations	□	□	□	□	□	□
Text Annotations	□	□	□	□	□	□

E2. What is your opinion on sharing metadata regarding experimental requirements?

	Not Important At All	Little Importance	Average Importance	Very Important	Absolutely Essential	Not applicable
Experiment Materials	□	□	□	□	□	□
Instruments/Devices Used	□	□	□	□	□	□

E3. What is your opinion on sharing metadata regarding settings?

	Not Important At All	Little Importance	Average Importance	Very Important	Absolutely Essential	Not applicable
Instrument Settings	□	□	□	□	□	□
Experiment Environment Conditions	□	□	□	□	□	□
Publications used	□	□	□	□	□	□

E4. What is your opinion on knowing the names and contacts of people/organizations who are involved directly (eg. Experimenter, Supervisor) or indirectly (eg. Manufacturer, Distributor) in your experiment and their roles?

	Not Important At All	Little Importance	Average Importance	Very Important	Absolutely Essential	Not applicable
Names of people who are directly involved	□	□	□	□	□	□
Contacts of people who are directly involved	□	□	□	□	□	□
Roles of people who are directly involved	□	□	□	□	□	□
Names of people who are indirectly involved	□	□	□	□	□	□
Contacts of people who are indirectly involved	□	□	□	□	□	□
Roles of people who are indirectly involved	□	□	□	□	□	□

E5. What is your opinion on sharing metadata regarding time, duration, and the location of experiments?

	Not Important At All	Little Importance	Average Importance	Very Important	Absolutely Essential	Not applicable
Date	□	□	□	□	□	□
Time	□	□	□	□	□	□
Duration	□	□	□	□	□	□
Location	□	□	□	□	□	□

E6. What is your opinion on sharing metadata regarding software used?

	Not Important At All	Little Importance	Average Importance	Very Important	Absolutely Essential	Not applicable
Software Parameters	□	□	□	□	□	□
Software Version	□	□	□	□	□	□
Software License	□	□	□	□	□	□
Scripts/Code/Program	□	□	□	□	□	□

E7. What is your opinion on sharing metadata regarding all the steps and plans?

	Not Important At All	Little Importance	Average Importance	Very Important	Absolutely Essential	Not applicable
Laboratory Protocols	□	□	□	□	□	□
Methods	□	□	□	□	□	□
Activities/Steps	□	□	□	□	□	□
Order of Activities/Steps	□	□	□	□	□	□
Validation Methods	□	□	□	□	□	□
Quality Control Methods	□	□	□	□	□	□

E8. What is your opinion on sharing the intermediate and final results of each trial of your experiments?

	Not Important At All	Little Importance	Average Importance	Very Important	Absolutely Essential	Not applicable
Final Results	□	□	□	□	□	□
Intermediate Results	□	□	□	□	□	□

E9. Please let us know what else should be shared when publishing experimental results.

Section F:

Experiment Workflow/Research Practices

In this section, we would like to know about your experiment workflow and your research practices.

F1. **What kind of data do you work primarily with?**

Images □

Multimedia files (Video, Audio) □

Measurements □

Graphs □

Tabular □

Other □

Other

F2. **Where do you store your experimental data files?**

Personal Devices (eg. Computer) □

Local Server provided at your workplace □

Removable Storage Device (eg. USB, Harddisk, CD Drive) □

Version Controlled Repositories (eg. Github, GitLab, Figshare, Zenodo etc.) □

Data Management Platforms □

Other □

Other

F3. **Where do you save your experimental metadata like descriptions of experiment, methods, samples used?**

	Primary Source	Secondary Source	Other
Hand written Lab Notebooks	□	□	□

	Primary Source	Secondary Source	Other
Electronic Notebooks	□	□	□
Data Management Platforms	□	□	□
Other	□	□	□

F4. Do you write scripts or program to perform data analysis at any stage in your experimental workflow?

Yes □

No □

Sometimes □

F5. Have you heard about the FAIR (Findable, Accessible, Interoperable, Reusable) principles?

Yes □

No □

Heard, but I don't know what exactly FAIR means □

F6. Does your research follow the FAIR (Findable, Accessible, Interoperable, Reusable) principles?

	Always	Often	Sometimes	Rarely	Never
Findable	□	□	□	□	□
Accessible	□	□	□	□	□
Interoperable	□	□	□	□	□
Reusable	□	□	□	□	□

F7. Please feel free to provide comments regarding what you think is important to enable understandability and reproducibility of scientific experiments in your field of research.

C.1 Survey Response for free text field questions

Please let us know what else should be shared when publishing experimental results.

1. The minimum information standards of the respective domains are a good starting point. I am generally in favor of open notebook science which aims to be totally open about everything as soon as the data, planning, etc is done.

2. Platforms should provide easy access.

3. Hidden parameters for data processing and reasoning for specific choices made for methods, steps, parameters.

4. The current academic rewarding system is pushing people into coming up with a nice story which unfortunately is encouraging people to publish their results without properly validating, hiding their negative data, adjusting statistical tests in a way that shows a significant difference and so on. The whole system is broken and has to change. Pre-registration of experimental plans, openly sharing lab notebooks, sharing all versions of the manuscripts along with reviewer's comments and answers to those comments, seperately publishing underlying datasets, codes and methods and therefore not forcing, polishing and hiding data to make a nice story but being open and transparent from the beginning and sharing all elements of research as individual items. To incentivize all these, promotion and hiring criteria should not only look for high impact journal publications but rather these type of efforts. Researchers typically spent the least effort to explain their materials and methods while that is one of the most important elements for research reproducibility. Dedicated methods repositories that archive not only the experimental procedures and parameters such as protocols.io but also videos of the procedures performed by the researchers would help enormously.

5. Metadata in a standardized format; License for data reuse

6. data owner (contact) property right

7. factors that negatively influence the outcome/working of an experiment supplementary results or negative results which might not be important for the published story may also be shared.

8. Ethics approval, the systematic review conducted before and alongside the study, the limitations, the contact people in project with long-term access. Everything should be shared in an structured findable, accessible, inter-operable, and reusable format (following FAIR guidelines).

9. A permanent ID for data with correpsonding license
10. Protocols used in the study with versions/adjustments made
11. URL/DOI Links to data in curated repositories; data availability statement
12. Computational environment needs to be fully specified, including OS and any software dependencies

Please feel free to provide comments regarding what you think is important to enable understandability and reproducibility of scientific experiments in your field of research.

1. The bottleneck for experimental scientists is that FAIR data sharing comes on top of everything else they have to do to generate and analyze the data. They are usually not experts in data handling/storage. The platforms we share our data on are often made by IT experts that do not realize that 'their language' and expertise is not immediately clear to biologists. It thus costs a lot of extra time and energy for experimental biologists to share their data. Moreover, there is still a feeling among my lab scientists that it is unfair that they are forced to share data, but that the one taking the data and doing synthesis projects (yielding high IF papers) never have to go in the lab and do the hard work of getting the data. We discussed this very often. Getting credits for sharing data does not sufficiently resolve this issue for them.

2. As a data manager I cannot really answer questions about the quality of my data, as I manage data of others and don't have own research data. I also cannot say where that data is saved as it always depends on the customer.

3. Internationally accepted metadata schemata covering all disciplines Controlled vocabularies covering most of the metadata fields An agreement on file formats

4. It is really important (in the case of hand-written notes) the scientists explains all the abbreviations used in her/ his lab books. It is also very important to keep the same structure of storing the experimental details/ steps (dates, treatments, titles).

5. Data sharing among other lab members is important to understand the reproducibility of the experiments.

6. I think it should be a criteria for research funders to allocate funding for reproducibility of each research and make it a mandatory criteria. Follow community/domain conventions. Eg when scripting in a particular language, follow software engineering conventions of that particular language to package up code.

Appendix D

CAESAR User Evaluation

D.1 CAESAR (CollAborative Environment for Scientific Analysis with Reproducibility) User Evaluation Responses

These are the responses of the participants of this study. The number in each row of the table denotes the number of participants who selected each category.

1. **Please rate the perceived usefulness of CAESAR.**

	Strongly Agree	**Agree**	**Neither agree nor disagree**	**Disagree**	**Strongly disagree**
It enables me to organize my experimental data more efficiently	2	4	0	0	0
Preserving data in CAESAR helps the new comers in the project to understand the ongoing work in the team	4	2	0	0	0
It helps me to search all the data related to my experiments including images, their metadata and device settings	4	2	0	0	0
It enables a collaborative environment among my team members	2	4	0	0	0
It enables me to visualize all the experimental data and results effectively	3	2	0	1	0
It enables me to link the images to the experimental data and results	2	4	0	0	0

2. **Please rate the following questions in regard to your experience with CAESAR.**

	Strongly Agree	**Agree**	**Neither agree nor disagree**	**Disagree**	**Strongly disagree**
CAESAR is useful for your scientific data management	2	3	0	0	0
CAESAR is user-friendly	0	3	2	0	0
CAESAR provides a collaborative environment among teams	0	5	0	0	0
It is easy to learn to use it	0	2	3	0	0

3. **What do you think about the following features in CAESAR?**

	Strongly Like	**Like**	**Neither like nor dislike**	**Dislike**	**Strongly dislike**
Project Dashboard (An one-place overview of all the experiments for a project)	1	3	1	0	0
ProvTrack (A visualization module to track the experimental data including the link between images, experiments and metadata)	3	2	0	0	0
ProvBook (A computational Reproducibility framework for data analysis scripts in Jupyter Notebook)	2	1	2	0	0

4. **Please let us know the overall feedback of CAESAR along with its positive aspects and the things to improve.**

- CAESAR provides a lot of features. Therefore, it is difficult to follow them.
- I find ProvTrack and ProvBook very useful among all the features in CAESAR. Sharing data among team members becomes easy with it.
- CAESAR has the potential to be a valuable addition to the “Materials & Methods” section of a scientific publication. It makes it easy to find the resources used in an experiment by simple“clicking” via the many connections between the elements in the database, so that it is much clearer how a measurement was produced. This is also very useful for the internal organization of a research group as CAESAR enables e.g. new lab members to get a better overview over the experimental workflow. The main issue is the stability of the connection to the server. This is especially the case with bigger files.