Reproducible Research and Interactive Data Mining in Bioinformatics.

A design science research in Biomedical Genetics

Master Thesis, Utrecht, 27th August 2015

Author
Armel E.J.L. Lefebvre - 4074483
Master Business Informatics Student
Utrecht University, Information & Computing Sciences
armelefebvre@gmail.com

1st Supervisor	***2nd Supervisor***	***External Supervisor***
Dr. Marco R. Spruit	*Drs. Ing. Wienand A. Omta*	*Dr. Wigard P. Kloosterman*
Utrecht University	*Utrecht University - UMCU*	*UMCU – Biomedical Genetics*

Universiteit Utrecht

UMC

University Medical Center
Utrecht

University Medical Center
Utrecht

Reproducible Research and Interactive Data Mining in Bioinformatics.

A design science research in Biomedical Genetics

Abstract

Data analysis of Next-Generation sequencing data is widely recognized as being a bottleneck on the way to understanding the human genome and personalize treatments. Studies argue for more integrative and interactive data analytics solutions that would largely automate and accelerate scientific discovery. At the same time, concerns are raised about the communication of computational experiments (CE) and their components which are code, data and algorithms. These concerns are mainly propagated by proponents of *Reproducible Research* (RR).

This research attempts to embed these interactive knowledge discovery practices with RR. This while investigating how RR constraints of sharing and reuse of data and code are applicable in real settings. To achieve this a prototype implementing the four steps of the HCI-KDD process (integration, preprocessing, mining and visualization/interaction) was developed and tested by biologists and bioinformaticians. The prototype is built around web resources to enable sharing and reuse of components produced during the KDD process.

Feedback from the prototype evaluation via three focus groups and one survey is summarized in a context enriched HCI-KDD process. This design proposition named RRO-KDD (Reproducible Resource-Oriented Knowledge discovery from data) merges HCI-KDD activities with design choices of the prototype. The goal is to improve reusability and sharing of previous work while taking into account how data is actually used and processed in biomedical research.

Our results suggest that there is room for improvement for applications enabling data analytics for biologists working with bioinformaticians. Sharing code and data as-is is not the optimal way to positively impact reuse of previous work as there is no sufficient contextual information to make retrieval convenient for both type of users involved in CEs. Web resources for visualization, data objects and research objects have the potential to be combined to address both the needs for interactivity and reusability better than the current practices suggested by Reproducible Research.

Acknowledgements

Many thanks to Wigard and his team. Wigard for giving the necessary help to build IT systems in a 150% biology concentrate environment. From explanations about what is hidden behind the data to contacts to get feedback about the “Wigard’s RNASeq Mining Tool” from Rotterdam. Also Alessio, Anne, Ivo, Myriam, Ronne, Mark and Mark for being present at the demo sessions leading to fruitful discussions.

To Marco and Wienand for letting me conduct research with a lot of freedom and make enriching experimentations with code and data. Especially Marco for offering the opportunity to get in touch with topics related to Life Sciences from the start of my master in Business Informatics.

I wish to thank also Peter de Waal and Ad Feelders. Peter for demonstrating that statistics are not as horrible as I ever thought and without this fear some papers were less terrifying to read. Ad for teaching the kind of data mining course that we really need before we are just even thinking about playing with data. The precious knowledge gained by learning from you made one of the most important aspects of this thesis manageable.

I would not have been able to start this master without the help of Jean-Marc Wagner (HEPL, Liege) and Nancy Roosens (WIV-ISP, Brussels) and the recommendation letters they wrote. Many thanks to Jean-Marc and the staff from Seraing for teaching me how to deal with computers in every situation. Thanks a lot Nancy (and PBB, Philippe and ICT) for everything I could do and learn at the Scientific Institute to push this knowledge to the next level.

And last but not least, my parents who I would have forgotten to thank in the acknowledgement section for the second time if I didn't do it now, and this would have been unfair!

Contents

List of figures

List of tables

Glossary

The definitions provided below only hold for this thesis and are not meant to be comprehensive.

Assumption Prior knowledge about a population (distribution, sampling) affecting the validity of statistical tests. In this work, the term assumption is generalized to any prior supposition about the state of the world when selecting or creating models to analyze data and draw conclusions.

Normalization Normalization of RNA-Seq counts data reduces noise and enables comparison between samples.

Pipeline A sequence of algorithms or tools applied on raw data to get analytical data

Representation A human or machine readable payload (information) of a *resource*. A resource may have one or more representation(s). These are, for instance, json, html, binaries…

Resource An object of interest (data, chart, method) accessible through a URI (*dereferenceable*).

REST API Querying resources through HTTP. Resources can be acquired, created, updated or deleted using HTTP verbs (GET, POST, UPDATE, DELETE).

RNA-Seq Acquire a snapshot of the transcriptome of some cells (tumor tissues/healthy tissues). One application is a comparison of expression levels between two treatments or tissues in one patient.

RNASeqTool Also RNASeq Mining tool, is the socio-technical artifact developed and deployed during the intervention phase of this design science research.

RRO-KDD Reproducible/Reusable Resource Oriented Knowledge Discovery from Data process is a design proposition resulting from the evaluation of the RNASeqTool and a literature review of Reproducible Research.

Sequencing Obtaining a sequence of nucleotides from a DNA fragment. A sequence is reported as an ordered list containing four characters (A, T, C, and G). Each corresponding to nucleotides.

Transcriptome In short, set of transcribed elements of the genome. Genome is DNA whereas transcriptome is RNA.

Table of counts Matrix of discrete data with genes in rows and samples in columns. Tables used in this work had in average 200 samples and 60000 genes.

Computational Experiment A CE is an experiment on data using tools or code (scripts, programs). They are the “virtual” counterpart of a lab experiment.

Chapter 1 Introduction

Data analysis of (large) biomedical data sets appears to be one of the biggest challenges in computational biology (Holzinger & Jurisica, 2014; Huang & Gottardo, 2013). Furthermore, the analyzed data sets are complex and require knowledge in bioinformatics, programming languages, (molecular) biology, machine learning, statistics and a large panel of lab instruments. Hence, collaboration between people and scientific disciplines is crucial.

In most cases, the data on which the actual analyses are based are an outcome of a chain of transformations (Calabria et al., 2014). The pipeline at the origin of the data for this thesis, for instance, is starting, from raw DNA/RNA sequencing data, going through an alignment to a reference genome and ending by aggregating, counting and normalizing data points (Calabria et al., 2014; Quackenbush, 2002). These interactions between computing systems impact how the interpreted files are generated. This fact combined with statistical or modeling assumptions inherent to algorithms and software packages used may induce significant differences in the data on which any further interpretation is based on.

Data mining, and to a larger extent the process of knowledge discovery in databases (KDD) is defined as “the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data” (Fayyad, Piatetsky-Shapiro, & Smyth, 1996) . Reproducible research invites researchers to share (meta)information with their peers and describe their intention and the methods applied for a given study. The first activity, data mining, is a well-established research area where literature is abundant, certainly in the bioinformatics field. The latter, reproducible research, ranges from (1) a battle against fraud in life sciences research (Laine, Goodman, Griswold, & Sox, 2007), (2) production of makefiles (Hoefling & Rossini, 2014), (3) design of Reproducible Research Systems (RRS) (Adhianto et al., 2010; Goecks, Nekrutenko, & Taylor, 2010) or (4) availability of components that constitute a computational experiment (R. C. Gentleman et al., 2004; R. Gentleman & Lang, 2007).

From the various goals, sometimes unconnected, of reproducible research (RR), one goal has to be chosen to ensure some consistency across this work. The selected aspect is close to fourth goal previously enumerated: making components of a computational experiment available. These components can be data transformation techniques, (un)supervised algorithms or plots with the common denominator of being interactively generated or triggered by a human user seeking to obtain new insights from his data.

First, we describe the high-level context of our research. Why are these data sets generated and for what kind of clinical research. More information can be found in Chapter 3 and Chapter 4 about, respectively, reproducible research for computational experiments and interactive data mining in genomics. An example of knowledge discovery for RNA-Seq Is given in Chapter 5. The implementation of a product artifact for knowledge discovery is provided in Chapter 6.

Next, Chapter 7 presents the feedback collected about the prototype. The output of the product artifact and the evaluation is presented in Chapter 8 where we introduce the lessons learned as a Reproducible Resource Oriented KDD process. This process links how the components were implemented and evaluated by experts or computer testing. The purpose of this process is to provide design guidelines to developers, bioinformaticians, computational biologists or lab managers. It aggregates software development practices and tools that positively impacts the reusability of their computational work. Finally, in Chapter 9 we conclude this thesis by addressing limitations and give some insight on future research.

1.1 Research context

1.1.1 Why so much data?

Bioinformatics, by designing or applying models on flows of data erupting from a wide diversity of lab instruments, is truly expected to be at the origin of major medical applications. Among these potential applications are new treatments against cancer or a better response to treatments (Chin, Andersen, & Futreal, 2011). As we may understand, these findings are supposed to be available to patients soon after their inception. So, in addition to producing a huge quantity of data, their processing should be accelerated in order to result in treatments tailored to patients faster.

This idea of clinical research directly pushed to therapeutics, also known as "from bench to bedside (b2b)" (Sarkar, 2010), is basically the high-level context in which our research will take place: translational medicine. Figure 1, based on Shortliffe & Cimino (2014), illustrates this concept and the role of bioinformatics (bench/research side) compared to health informatics (patient side). Translation in this context implies the transfer of the outcomes of bioinformatics to human health.

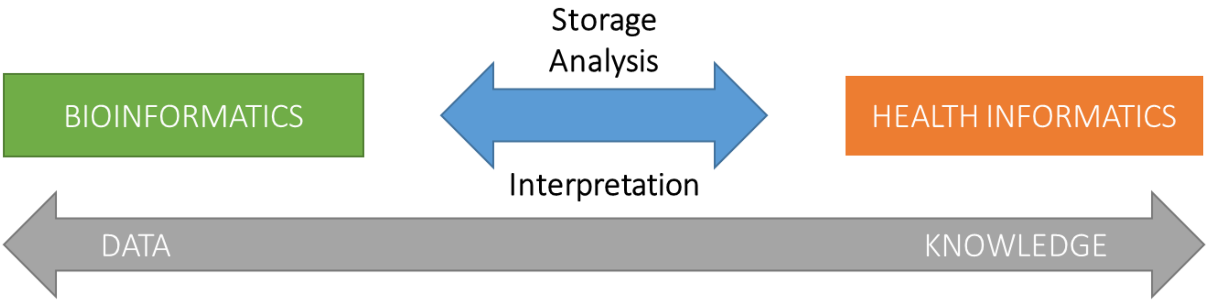


Figure 1 Illustration of the translational process based on *Shortliffe & Cimino (2014)*

Data is in this case related to the genome of a patient. There is a constant struggle to make full genome sequencing less expensive. While some technologies defend a lower sequencing cost a second facet is its analysis which is not as straightforward to budget (see next section 1.1.2).

1.1.2 The $1.000 genome… at $1 Million interpretation?

This misbalance between data generation and data analysis expressed in dollars was nicely put in words by Bruce Korf (ACMG) in 2010. In terms of expenses, the goal is set toward the $1000 genome by, for instance, sharing DNA sequencing platforms (outsourcing). The second aspect, its interpretation is globally recognized as the bottleneck (Chin et al., 2011; Scholz, Lo, & Chain, 2012).

This fact leads to asking ourselves how to facilitate the collaboration between bioinformaticians and biologists? How could the practices defended by the reproducible research movement help us to design systems making this knowledge and interpretation easier?

As we have observed during the development of a socio-technical artifact, biologists are also coding scripts, which does not leave the computational part of an experiment to bioinformatics only. There is room for a better integration of computational experiments and pipelines as managed by bioinformaticians and smaller scripts written by biologists.

1.2 Scientific and societal relevance

1.2.1 Scientific relevance

The project is focusing on two aspects of data analysis in the biomedical genetics field. Interactive data mining (IDM) and reproducible research (RR). First, interactive data mining methods and frameworks are often advocated as a solution for empowering domain experts in analyzing complex data. By leveraging visualization tools combined with human cognition (known as human-computer interaction, HCI), knowledge creation and sharing are not expected to be left to computers alone. For instance, the HCI-KDD process suggested by Holzinger & Jurisica (2014) aims at making the KDD process more interactive for researchers. Reproducibility is emphasized as a crucial element but not thoroughly investigated in Holzinger, Dehmer, & Jurisica (2014). Therefore, we consider that bringing the extra dimension of reproducibility to computational experiments conducted by biologists and bioinformaticians needs to be investigated further.

1.2.2 Social relevance

Enhancing the knowledge discovery experience with components capturing the ideas, hypotheses and footprints of data analyses will be beneficial to safely conduct translational research in biomedical sciences. The overall goal is to improve the elaboration of treatments targeted to patients as prescribed by the personalized medicine approach (Hamburg & Collins, 2010). This is based on claims of authors who relate an increased trust in therapies to an application of a *reproducible research* paradigm by researchers (Laine et al., 2007).

There are numerous aspects of linking computational experiments to reproducible research practices. Here, we consider mainly technical aspects of *reproducible* knowledge discovery in databases. What is concluded from the literature is that in order to be maximally impacting daily practice in research, organizational changes are also desired. First, that publishers encourage such practices by promoting a “culture of reproducibility” and reflect on new ways of communicating knowledge and discoveries (Moseley, Hsu, Stone, & Celi, 2014; Peng, 2011). Second, that best practices in IT are taught to biologists or bioinformaticians that are coding scripts without being aware of tools (versioning, IDEs…) that are commonly used by professional developers (Sandve, Nekrutenko, Taylor, & Hovig, 2013).

Recently, journals are putting forward these principles also to counter fraud in science by enabling a better verification by reusing the computational components of published studies. This appears in McNutt (2014), for instance, with a focus on preclinical research.

1.3 Problem statement

Biologists seek to analyze and visualize their data sets with interactive end-user interfaces. These data sets are the result of one or more bioinformatics pipelines with specificities that may be unknown to these end-users. Meanwhile, publishers are asking for more detailed information about the methods or data processed during a study. Hence, biologists and bioinformaticians have to collaborate to pass the analysis challenges and opportunities offered by sequencing technologies (next-generation and third-generation).

> *The issue of reproducibility of computational experiments is focused on code and data availability. At the same time, more interactive tools are designed for end-users who are not coding. There is little knowledge about how to guarantee reproducibility of CEs conducted via interactive interfaces. Moreover, the extent to which reproducibility is applicable or important in daily practices is not extensively known.*

This leads to possible improvements in how bioinformatics and biology researchers collaborate on their data analytics by having more information about how the data sets that are processed, analyzed and interpreted and hence gain a better ability to share their results.

1.4 Research questions

Following the problem statement, a main research question is divided into 5 research questions.

> *MQ: "How to ensure that a computational experiment conducted by means of an interactive knowledge discovery process using web resources and technologies offers an adequate level of reproducibility?"*

SQ1.1:"What is reproducible research for computational experiments, how is it defined and what existing methods and tools support it and how?" (Chapter 3)

SQ1.2:"What aspects are relevant to communicate along with results of an analysis and how to make a computational experiment accessible to the research community?" (Chapter 3)

SQ1.3:"What are the data (pre)processing and visualization techniques that are relevant to interactively explore gene expression RNA-Seq data and visualize outliers?" (Chapter 4)

SQ1.4:"What are the functional and technical components of a web application that enable or limit reproducible data (pre)processing, interpretation and exchange?" (Chapter 6 and Chapter 7).

SQ1.5: "What are the characteristics that have to be implemented in a knowledge discovery process to enable reproducible data (pre)processing, interpretation and exchange with the research community?" (Chapter 8)

Chapter 2 Design science research

A design science research was applied during this thesis project. There are two deliverables which are a socio-technical artifact (a web application) and a process artifact (knowledge discovery with resources). The fact that the first artifact is *socio-technical* has some implications for communication and evaluation that will be explained later in this chapter. The main reason is that we try to capture knowledge about the field of biomedical genetics and to act on it with the medium of a tool. It is therefore a type of artifact which aims at *transforming* the state of the world instead of describing it and there is a large uncertainty in methods (naturalistic or artificial) to assess or demonstrate the influence of such artifacts (Carlsson, Henningson, Hrastinski, & Keller, 2011). Instead of being sunk in endless debates, naturalistic evaluation methods were selected as they let specialists speak their minds around a tangible IT artifact[1]. Figure 2 summaries the high-level tasks of design science that were applied during this research.

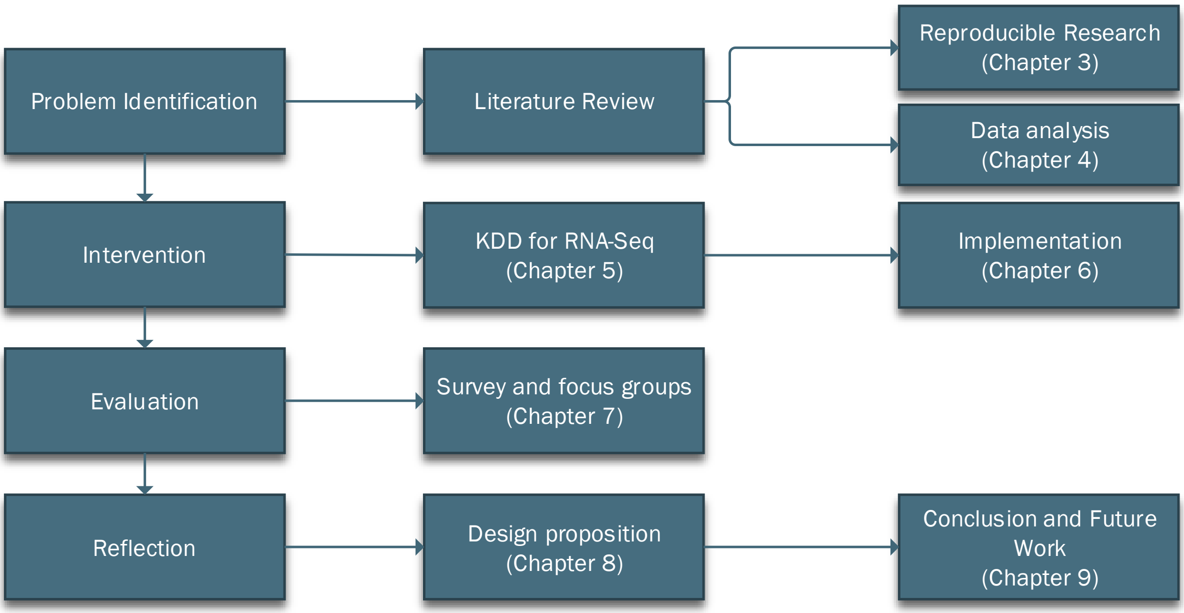

Problem Identification
Literature Review
Reproducible Research (Chapter 3)
Data analysis (Chapter 4)
Intervention
KDD for RNA-Seq (Chapter 5)
Implementation (Chapter 6)
Evaluation
Survey and focus groups (Chapter 7)
Reflection
Design proposition (Chapter 8)
Conclusion and Future Work (Chapter 9)

Figure 2 Overview of the main components of the approach by Hevner & Chatterjee (2010)

Most parts of the design science research approach are following the guidelines of Hevner & Chatterjee (2010). First, *problem definition* was addressed in sections 1.3 and 1.4 with a problem statement and research questions. Second, *intervention* was the creation of a web application (commonly named RNASeqTool) with weekly meetings, exploratory focus groups or "ad-hoc" meetings made possible by the availability of bioinformaticians or biologists. Third, the socio-technical artifact was evaluated for the design choices and somewhat its usability to get even more ways to improve the prototype. Finally, the second artifact is a formalization of the tool with extension to theoretical and technical concepts that might improve reproducibility of computational experiments (CE) in a context of interactive mining done by biologists. Figure 3 illustrates the DSR framework adapted to this study.

[1] And the fact that objectively tracking eye-rolling in front of usability misconceptions or brain activity during outlier streaming was totally out of the scope of this thesis.

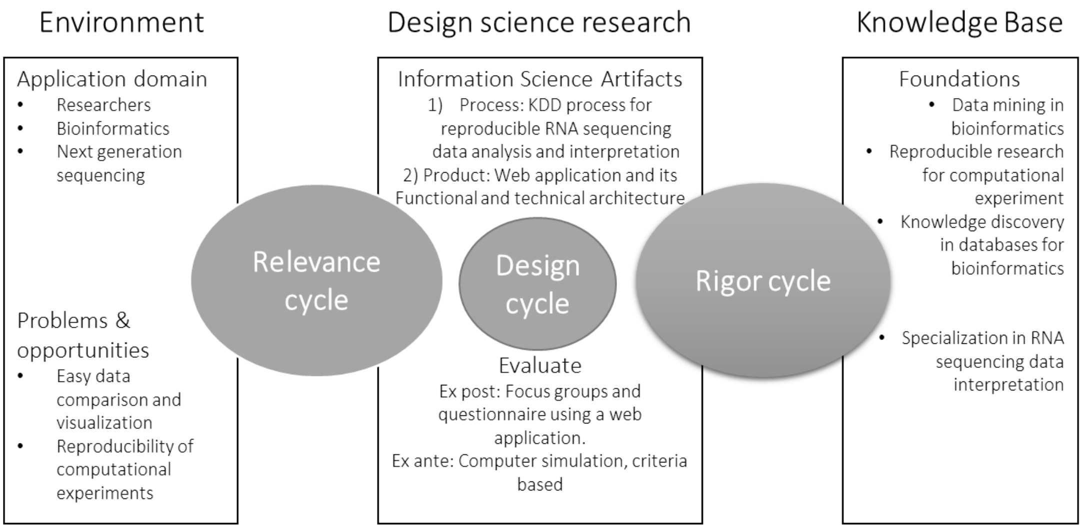

Environment
Design science research
Knowledge Base
Application domain
• Researchers
• Bioinformatics
• Next generation sequencing
Problems & opportunities
• Easy data comparison and visualization
• Reproducibility of computational experiments
Information Science Artifacts
1) Process: KDD process for reproducible RNA sequencing data analysis and interpretation
2) Product: Web application and its Functional and technical architecture
Relevance cycle
Design cycle
Rigor cycle
Evaluate
Ex post: Focus groups and questionnaire using a web application.
Ex ante: Computer simulation, criteria based
Foundations
• Data mining in bioinformatics
• Reproducible research for computational experiment
• Knowledge discovery in databases for bioinformatics
• Specialization in RNA sequencing data interpretation

Figure 3 Design research framework adapted to our case

2.1 Artifacts in design science

This section is a checklist to ensure that we comply with the seven criteria of DSR as formulated by Hevner & Chatterjee (2010). We also shortly review and describe which artifacts were created and evaluated during this project.

CRITERION	JUSTIFICATION
DESIGN AS AN ARTIFACT	A product artifact consisting of a web application and an API to perform analysis of gene counts (matrix) generated by RNA-Seq data. A process artifact is a Reproducible Resource-Oriented Knowledge Discovery Process (RRO-KDD) reflecting the methods and techniques implemented in the product artifact. The RRO-KDD is fundamentally a “lessons learned” and suggestions to design a knowledge discovery process with reproducibility in mind.
PROBLEM RELEVANCE	Both problems regarding data analytics of next generation sequencing techniques and how to keep track and retrieve elements of these analyses are highly relevant to the fields of bioinformatics and information science.
DESIGN EVALUATION	The technical artifact is evaluated by experts (biologists and bioinformaticians). It serves as a knowledge base (or even pump) to get the most realistic overview of what’s happening in the field.
RESEARCH CONTRIBUTION	This research aims at contributing to the creation of novel techniques (Virtual experiments and Research Objects) supported by the web architecture to disseminate knowledge from exploratory data analyses performed in an interactive manner (for end-users that are not hardcore programmers) in bioinformatics.
RESEARCH RIGOR	The implementation of a working tool and the present thesis document serve as a way to evaluate the approach and theoretical foundations of the research project
DESIGN AS A SEARCH PROCESS	Iterations, demos, ad hoc meetings and literature reviews lead to many adaptations and form an outcome of a cyclical search process.

COMMUNICATION OF RESEARCH	Source code, presentations (MBI Colloquium and at the UMCU), a thesis document and a scientific paper are the main aspects of how this research is communicated.

Table 1 Design science principles guiding this research

2.1.1 Socio-technical artifact

The developed prototype is our custom measure instrument to get knowledge from the field about how data is analyzed, what are the advantages or problems with “self-service” bioinformatics that may influence the reproducibility of computational experiments. It is a working prototype, which proposes a set of features to analyze and visualize data together with some design choices linked to the fact that exchangeable footprints must be produced.

Sometimes inelegant design choices (like the strict separation between data manipulation and visualization) were discussed and processed with biologists and bioinformaticians. The last iteration of the tool suggest numerous ways to achieve a better design of research objects that are also suited to the hard demand of data visualization (which is dependent on all the previous workflows) and scenario analysis which is the dreamed feature. The goal of scenario analysis would be to easily play with different outputs of R packages or python modules (without coding) and *see* what happens with the samples or features. It is an unachieved requirement that still has all the reasons to be worked on.

2.1.2 Process artifact

It is a documented knowledge discovery process built upon the information gained by the evaluation of the tool and the research practices in Life sciences and more precisely in Next-Gen sequencing (NGS) for medical purposes.

It is presented as a process-deliverable diagram, a modeling technique elaborated in the field of method engineering (Brinkkemper, 1996). It offers an easy way to show how the knowledge discovery process generates outputs, what they expect to obtain at each step and what kind of information or documents could be provided.

2.2 Communication

2.2.1 Ad hoc meetings with experts

While the prototype was developed, an eye was kept on requirements or needs that practitioners would express. Mainly the Daily supervisor of this project, who is a biologist, introduced the sequencing techniques (RNA-Seq). Secondly, a bio-informatician of his team generated raw and normalized data sets of gene expression counts. First from samples of UMC Utrecht patients then extended to publicly available data sets (called GENENTECH and TCGA). Based on these existing files, possible data mining methods and visualizations were discussed but without a clear vision of what was needed and how accurate were some measurements in these files.

Usually, these meetings were followed up by a literature review and testing of R or Python packages/modules for data analysis and visualization. Static graphs and plots were also generated using various techniques and submitted to the professionals, firstly to judge how informative they were regarding their capability to give insight on underlying biological phenomena. Mostly, such plots were not enlightening or their biological accuracy was practically impossible to judge (distance/clustering between samples on a two-dimensional plot obtained by PCA or Factor analysis, for instance). As we will see, this was mainly due to hindrances created by the normalization methods applied and the nature of the data (number of normal samples versus tumor samples was largely imbalanced in the main data set used for this enquiry).

Nevertheless, these meetings (scheduled or not) were important to remain synchronized with the practitioners, refine problems and discuss potential solutions while implementing the prototype. This guarantees also that people contacted to evaluate the artifacts (KDD process and tool) will be familiar with the terms used and that no irrelevant techniques are actually added. Without entering into details, any software engineering oriented mind identifies the “agile” approach followed to realize a working prototype (Fowler & Highsmith, 2001).

Additionally, this development phase was conducted without complex requirement management processes in place. User requirements were not the one and only input for development as the focus was on how to make computational experiments reproducible. But it was also agreed upon that the web tool should be useful for practitioners and deployed in production after the research project. The reproducible aspect was entirely based on literature and observation of analyses in bioinformatics and was not an initial demand of the practitioners. It was simply confirmed by their own experience that, indeed, using data and methods from previous studies is a challenge even if data sets are publicly available.

2.2.2 Weekly presentation meetings

The department of Medical Genetics at the UMCU organizes work meetings every week. One or two presenters update the audience about their ongoing research. Assisting to these meetings provided additional insights on the problematic of working with data in Life Sciences but on a wide range of technologies or disciplines.

Precious implicit knowledge was gained to inspire a more robust and generalizable architecture based on the experience of people involved in diagnostics (patient-side) and research. It also illustrates the challenges and collaborations that occur and the role of data which appears to be unavoidable.

2.2.3 Exploratory focus groups

Additionally, four exploratory focus groups to discuss the implementation of the product artifact were organized. These focus groups started with a suggested architecture (user interface and WEB API) then a discussion about the data normalization techniques (considered as having a big impact on the rest of the data analysis) and what kind of visualization is suitable for an interactive data mining tool for mining outliers.

DATE	TOPIC	WHO
11/12	Introduction and architecture suggestion	Kloosterman group
05/02	Influence of methods on the data analysis, refinement of the main functionalities	Kloosterman group
13/03	Overview of normalization methods, presentation of the architecture (end-user interface and Web API)	Kloosterman group + First supervisor + second supervisor
09/04	Presentation demo	Kloosterman group + second supervisor

Table 2 Summary of exploratory focus groups

2.3 Literature reviews

A literature review was conducted at the early stage of this thesis project. It was focused on Human-Computer Interaction and knowledge discovery, ex. HCI-KDD (Holzinger, 2013; Jurisica, 2014), and reproducible research which was, at that time, mainly discussing how to log an entire computational experiment. This literature review was not systematic and conducted by “domain” of research with an attention to knowledge that may be overlapping between fields. Papers were found by snowballing and/or suggestions. Due to the broad and exploratory nature of this work, the literature review had to catch-up quickly with the state-of-the art techniques in IT and Biology to identify how most updated practices in IT can support the most current issues in Life Sciences and Cancer research.

For the more statistical/analysis part, the literature review started with papers suggested for each package (in Bioconductor for instance) and papers discussing these techniques were identified by the "referenced by" alike features of search engines.

After the project started, other domains had to be continuously investigated:

- Statistical analysis of RNA-Seq data: Basically, the prototype is built around different normalization methods that were identified in the literature. Normalization is a crucial step in differential gene expression analysis and also in outlier detection as the choice of a normalization methods impacts the scale of the data (which may be transformed from discrete counts to continuous values).
- Reproducible Research: Reproducible Research started with the idea to log every detail of a computational experiment in order to be able to fully repeat it by sharing code, data and extra details (like supplementary material in papers). First, these practices are certainly not "harmful" but are concentrated on the bioinformatics side of computational analysis and not on the more interactive, biology, side. Second, packaging the experiment might hinder its reusability by being tailored to the analysis of the provided data (strong coupling) and put the focus too much on the black-boxes (packages) instead of the core of experiments with data: statistical models, patterns…
- Technical field: to build the interface and to make it compliant with nowadays web development technologies, diverse areas were investigated.
  - Visualization with transformers: the plots are generated server side and sent to the clients as JavaScript and HTML chunks. This automated conversion from static (python or R) plots that are rendered dynamically to client avoid to inadvertently change the aspect, scales or axes for visualization and appears to be a good choice for "Reproducible" interactive plotting.
  - JavaScript: The website is clearly the interactive, user-friendly part of the system. Asynchronous calls, dynamic data-binding and project separation with the model-view-controller (MVC) pattern are guaranteed by Angular.js with additional plugins.
  - REST API: This is the hidden aspect to the end-user that is enabling the potential diffusion of resources created during the "interactive" data mining session via the website. They are implemented mostly as "plain old" Json messages but could potentially be changed in "Research objects" for internal or external reuse.

The outcomes of the literature reviews are disseminated in Chapter 3, Chapter 4 and Chapter 5.

2.4 Evaluation

2.4.1 A word about evaluating socio-technical artifacts in IS design science research

The evaluation steps in design science research are still a debated topic where no strong agreement on how to assess artifacts emerges (Pries-Heje, Baskerville, & Venable, 2008). Hence, the importance of this step to validate the contribution of the artifacts to a knowledge corpus is weakened by the absence of clear guidelines per type of artifact. According to Mettler, Eurich, & Winter (2014) the evaluation step is about (1) "Testing solution against requirements and (2) "Assessing impact in real world". When tested in the field, this evaluation procedure is defined as *naturalistic evaluation* which "explores the performance of a solution technology in its real environment i.e., within the organization." (Pries-Heje et al., 2008). There are two type of evaluations that were conducted:

2.4.2 *Ex ante*: Computer simulation and criteria-based analysis

The ex-ante evaluation takes place before experts manipulate the artifacts. Here, it was conducted by the author of this work based on computer simulations (testing, running scripts on example data sets) and criteria based assessments to filter out unrealistic or irrelevant components found in the literature or technical documentations of packages. The goal is to offer a subset of functionalities/possibilities to experts during their (ex-post) evaluation while being able to justify why some features were implemented and why some were not. It is also described as an appropriate strategy for providing early rationales that may be valuable to communicate to stakeholders in a project, e.g. researchers (Sonnenberg & Brocke, 2012).

2.4.3 *Ex post*: Applicability check: Focus groups and questionnaire

Once the product artifact (application) is developed, it can be tested by end-users and their feedback is collected in order to document a «Reproducible »Knowledge discovery process with the information perceived as important for experts (bioinformaticians and biologists). The focus groups were organized by inviting specialists that had no previous experience with the project settings (requirements, aim of the prototype, earlier discussions) except for the pre-test session. The evaluation with focus groups is elaborated by Tremblay, Hevner, & Berndt (2010) and this approach was implemented for our focus groups.

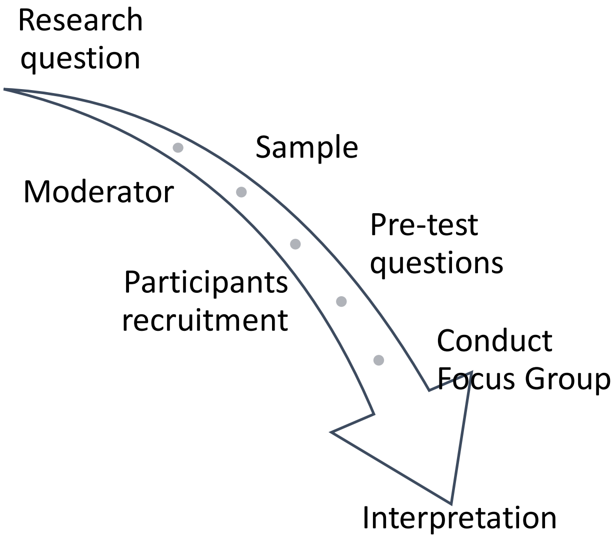


Figure 4 Steps to organize focus groups

The qualitative results of the focus groups are compared together (pre-test, internal and external) and a survey brings some quantitative support or orientation to the enhancement of the artifacts. The survey was a google form and a link to the form was posted on the test website. The survey was secondary to focus groups for the evaluation and provision of feedback.

2.5 Process-Deliverable Diagram of this work

In Figure 5, we use a process-deliverable diagram showing the main stages of this thesis.

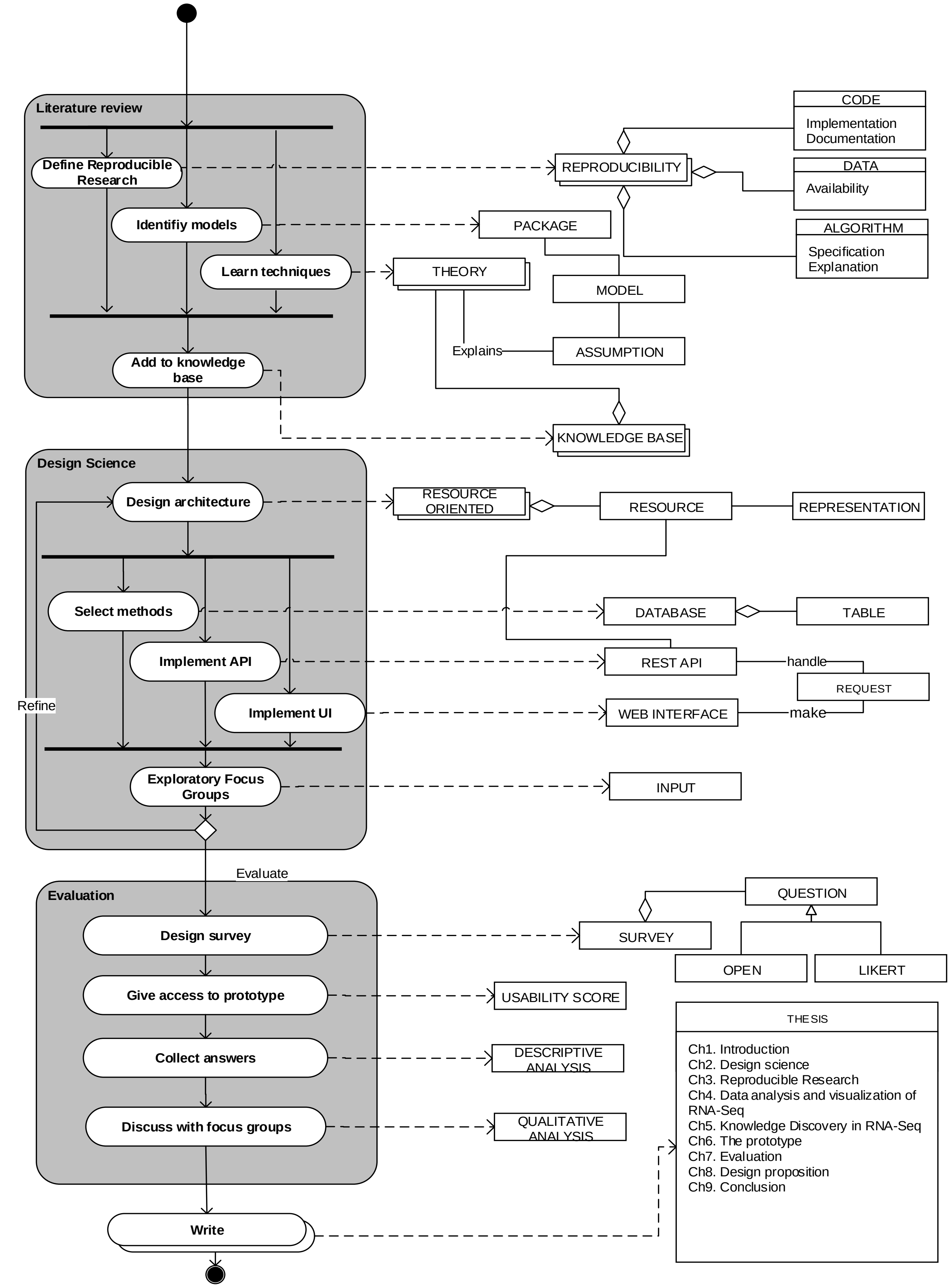


Figure 5Process-deliverable diagram of this thesis

Chapter 3 Reproducible research for computational experiments

In this chapter, we present some concepts and technical implementations for reproducible research. We limit ourselves to Reproducible Research tools that are mixing context and implementation.

3.1 Theoretical background

3.1.1 How to be Reproducible?

For years, reproducible research (RR) appears to have been underestimated in computational biology (Donoho, 2010) and recently emerged as a main concern illustrated by new research communities like the open science Framework (https://osf.io/s9tya/) or recent papers and publishers defining and encouraging RR (McNutt, 2014; Sandve et al., 2013). Reproducible Research Systems (RRS) like Galaxy (Goecks et al., 2010) and GenePattern (Kuehn, Liberzon, Reich, & Mesirov, 2008; Reich et al., 2006) also promote traceable pipelines and analyses in bioinformatics. Not only in the perspective of attaching relevant code and data to publications in the biomedical field (Peng, 2011) but also when considering the translational research context where *in silico* findings are expected to lead to clinical applications faster. As an example of the role of reproducible research in that case is a strongly encouraged ability to *reproduce results* related to published new biomarkers for cancer screening (Wagner & Srivastava, 2012).

It is also relevant to note that lack of reproducibility is not an unknown phenomenon at the biomedical genetics department. Indeed, the data used for building our interactive data mining tool is linked to a project seeking to identify recurrent fusion genes in colorectal tumors. Attempts have been made *in-house*, at the UMCU, to reproduce the outcome of a previous study (Seshagiri et al., 2012) which identified such genes. By investigating a downloaded data set provided by the authors, not all fusion genes listed in the study could be found in the data set except the recurrent fusion genes.

To begin with, we provide a summarizing table isolated from two papers of the “Ten simple rules” series published in PLoS. Figure 6 shows ten rules for reproducible (first column) and effective (second column) computational experiments. Despite the fact that those rules are meant as a recall of best practices to a large audience rather than an investigation of what makes a piece of software reproducible, we note that some of these rules are redundant across both dimensions (reproducibility and effectiveness). In addition, the reproducible and effective dimensions highlight understanding (context), tracking (version control) and sharing (public access) in one or more rules.

RULE	REPRODUCIBLE (SANDVE ET AL., 2013)	EFFECTIVE (OSBORNE ET AL., 2014)
1	For Every Result, Keep Track of How It Was Produced	Look Before You Leap
2	Avoid Manual Data Manipulation Steps	Develop a Prototype First
3	Archive the Exact Versions of All External Programs Used	Make Your Code Understandable to Others (and Yourself)
3	Version Control All Custom Scripts	Don't Underestimate the Complexity of Your Task
5	Record All Intermediate Results, When Possible in Standardized Formats	Understand the Mathematical, Numerical, and Computational Methods Underpinning Your Work
6	For Analyses That Include Randomness, Note Underlying Random Seeds	Use Pictures: They Really Are Worth a Thousand Words
7	Always Store Raw Data behind Plots	Version Control Everything

8	Generate Hierarchical Analysis Output, Allowing Layers of Increasing Detail to Be Inspected	Test Everything
9	Connect Textual Statements to Underlying Results	Share Everything
10	Provide Public Access to Scripts, Runs, and Results	Keep Going!

Figure 6 Ten rules for reproducible (Sandve et al., 2013) *or effective* (Osborne et al., 2014) *CEs.*

There are many definitions coined by different authors and the richness of their interpretations makes it harder to simply give consistency to the concept of *reproducible research*… The first issue being the differences between what is called replication and reproducibility. As we will see, there are some inconsistencies between authors to define if reproducible is a braindead version of replication (push on a button and get the same results) or the other way around like Drummond (2009).

Vandewalle, Kovacevic, & Vetterli (2009) defines reproducibility pretty much the same way as Wikipedia does[2] as they refer to the online encyclopedia: "reproducibility is one of the main principles of the scientific method and refers to the ability of a test or experiment to be accurately reproduced, or replicated, by someone else working independently".

Earlier, King referred to a *replication standard* as being the fact that "sufficient information exists with which to understand, evaluate, and build upon a prior work if a third party could replicate the results without any additional information from the author" (King, 1995, p. 444). After these two definitions, we might conclude that *replication replicates* and *reproducibility* is an *ability* to *replicate* or *reproduce*. It does not clarify the situation but additional information can be drawn. Both address the issue of an independent person/team who would has *sufficient information (or ability)* to process results based on previous findings… This sufficient information appears in our main research question under the term *adequate level*. The points which remain fuzzy is if reproducibility is "accurate" or an "alternative investigation"… Hereafter, *level* can be understood as "to what extent the information given is *sufficient* for biologists without overloading their computational investigation with thousands of parameters about instruments or algorithms".

Therefore a more practical sense of *reproducibility* is used in this thesis as a bootstrap to reproducible research. It is focused more on the problem of *sufficient information*. Peng, Dominici, & Zeger (2006) highlight already the complexity of analysis and processing of data sets, here in epidemiology. They divide data in two groups: *analytical data* and *measured data*. *Analytical* data is the data on which statistical analyses are done (e.g. a table of read counts in RNA-Seq analysis). Measured data is the processed data to generate analytical data (e.g. pipeline and tools counting reads per feature). They suggest that analytical data is made available (at least) as a requirement for Reproducible Research.

RESEARCH COMPONENT	**REQUIREMENT**
DATA	"Analytical data set is available." (Peng et al., 2006)
METHODS	"Computer code underlying figures, tables, and other principal results is made available in a human-readable form. In addition, the software environment necessary to execute that code is available." (Peng et al., 2006)

[2] Definition at http://en.wikipedia.org/w/index.php?title=Reproducibility&oldid=262130461

DOCUMENTATION	"Adequate documentation of the computer code, software environment, and analytical data set is available to enable others to repeat the analyses and to conduct other similar ones." (Peng et al., 2006)
DISTRIBUTION	"Standard methods of distribution are used for others to access the software, data, and documentation." (Peng et al., 2006)

Figure 7 Basic criteria for reproducibility (Peng et al., 2006)

Then, Roger D. Peng (2011), based on results obtained in a previous research on reproducibility of microarray gene expression analysis studies which yielded three categories (not reproducible, partially reproducible and fully reproducible), describes the *full spectrum of reproducibility* (see Figure 8). Surprisingly, while this paper was published later than (Peng et al., 2006), the full spectrum is less richer than the minimal requirements stated in Figure 7. Indeed, the notions of methods and documentation do not appear. Additionally, Peng notes that exploratory data analysis tools are often not designed for Reproducible Research and the fact that close-sourced systems may not put reproducibility as one of their quality attributes leads to a lack of evolvements (updates) in that way.

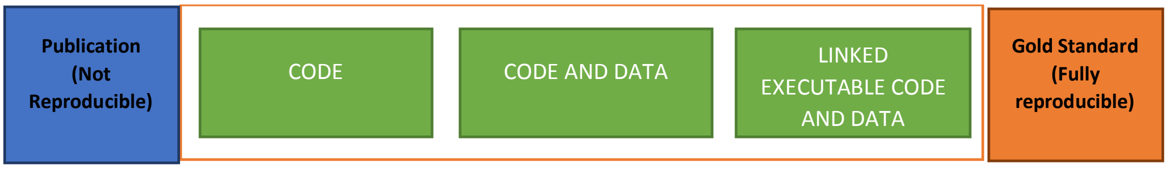

Publication (Not Reproducible)
CODE
CODE AND DATA
LINKED EXECUTABLE CODE AND DATA
Gold Standard (Fully reproducible)

Figure 8 The gold standard (Peng, 2011)

As can be seen from Figure 8, the emphasis lays on sharing *executable* code and data. To achieve this, some of the tools presented in the next section might be handy but we might ask ourselves if this is as easy (just share) to attain the gold standard. To be the gold standard, in that case, is when an independent team would need executable code and data as *sufficient information*.

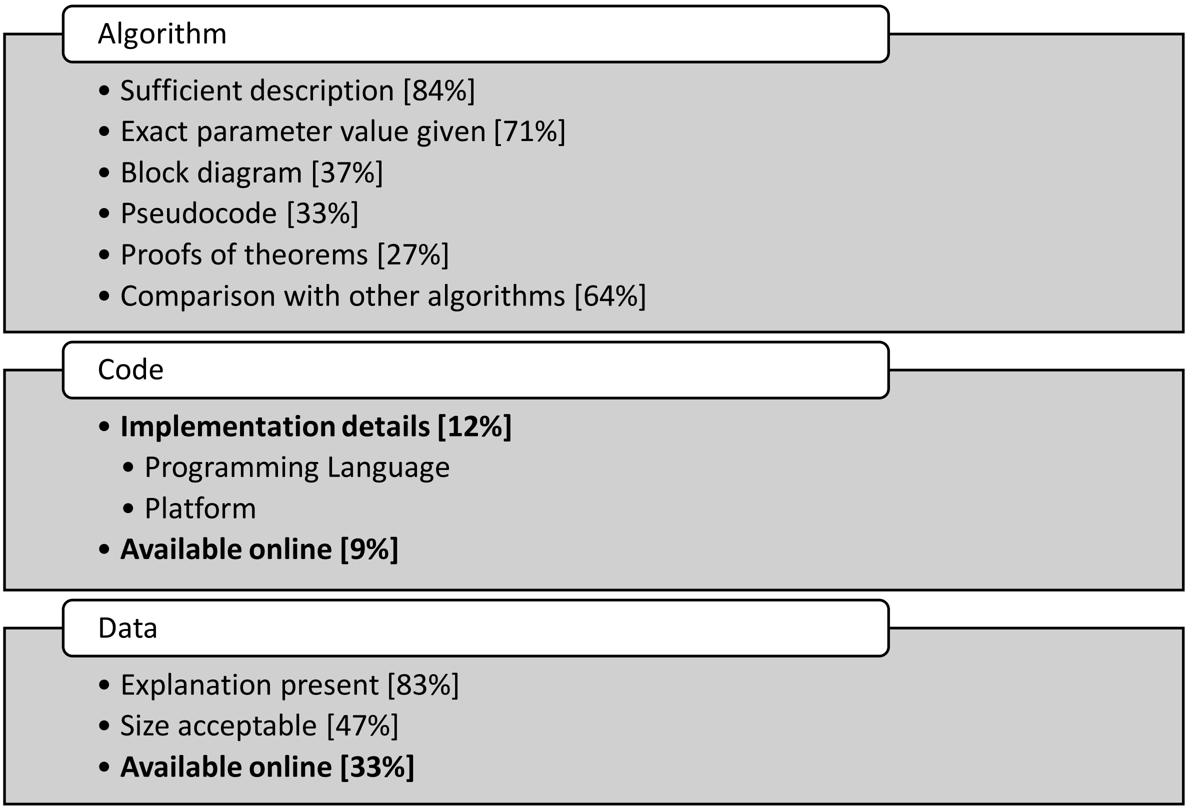
Meanwhile, Drummond (2009) brings an interesting addition to the understanding of reproducibility and *replicability*. The latter, according to Drummond, aims at "reproducing exactly" whereas reproducibility may yield similar results but through alternative experiments and thus "requires changes" (Drummond, 2009) and is a richer concept than replicability. What is pointed out is that the *verification* of an experiment through simply rerunning the code that others created is not "enough" to guarantee a reproducible experiment. Hence, according to this view, Davison & Mattioni (2014) with Sumatra, a tool which is designed to "keep track of all the experimental details: the scientist's own code, input and output data, supporting software, the computer hardware used, etc." are closer to a replicability scenario than to reproducibility, even if presented as primary for *reproducible research*. That means that they subscribe to the "record everything" view of reproducibility (i.e. replicability) in contrast to Drummond (2009).

Additionally, a list of advantages of reproducible research internal to an organization is offered by Donoho (2010, p. 386) are "(1) the ability to reuse methods developed internally, (2)Improve the way of working, (3) Improve the work as a team and (4) greater continuity, i.e. the training of new team members on previous work".

These advantages also shows that one important goal of RR is to be beneficial for researchers in the same lab, not only to verify or reuse work of third-parties but reuse the work of previous members (e.g. due to a high " turnover").

3.1.2 What do studies measure when judging reproducibility of previous work?

To get back to more concrete notions of Reproducible Research for our computational experiments, let us take a look at the state of code and data sharing in two domains: signal processing and bioinformatics. First with Vandewalle et al. (2009) that conducted an analysis on papers published in signal processing (n=134). Figure 9 summarizes the criteria used with corresponding results of their study. As can be seen, three categories are put forward: Algorithm, Code and Data. We see that for algorithms it is mostly formal proofs or a higher level description (block diagram or pseudo code) that are missing.



Algorithm

- Sufficient description [84%]
- Exact parameter value given [71%]
- Block diagram [37%]
- Pseudocode [33%]
- Proofs of theorems [27%]
- Comparison with other algorithms [64%]

Code

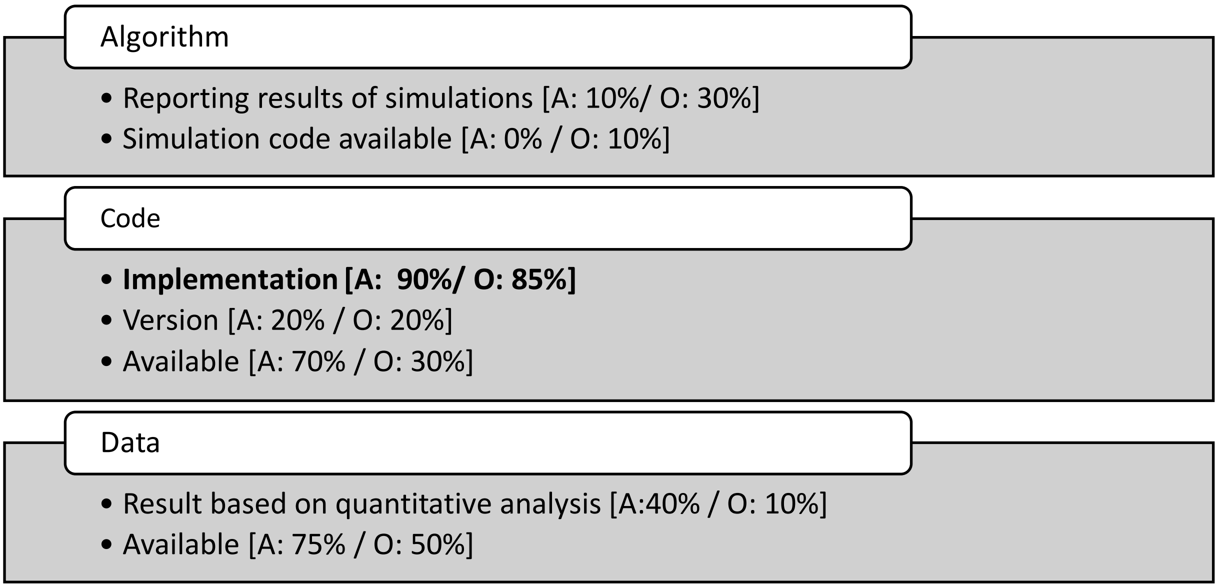
- **Implementation details [12%]**
  - Programming Language
  - Platform
- **Available online [9%]**

Data

- Explanation present [83%]
- Size acceptable [47%]
- **Available online [33%]**

Figure 9 Checking the " reproducibility" of a paper, criteria from (Vandewalle et al., 2009)

Later, a similar comparison is made by Hothorn & Leisch (2011) on papers published in *Bioinformatics* (n=100) randomly selected papers in Volume 26(1 – 7). The criteria differ slightly from the previous study (Vandewalle et al., 2009) and they distinct so-called original papers and application notes which are shorter communications about (new) software implementations. We grouped the criteria from (Hothorn & Leisch, 2011) in identical sections as Vandewalle et al. (2009) to make the comparison between two studies easier to interpret although there is no exact match between the categories and criteria applied. The outcome is provided below in Figure 10, the percentage given mixes the answers Yes and "Available upon request" if applicable. Here there is a distinction between code availability of simulations and of software used that is not really clear from the paper. The mean of both type of manuscripts and both categories is 27.5% of code availability, which is still three-fold higher than signal processing (9%):



Algorithm

- Reporting results of simulations [A: 10%/ O: 30%]
- Simulation code available [A: 0% / O: 10%]

Code

- **Implementation [A: 90%/ O: 85%]**
- Version [A: 20% / O: 20%]
- Available [A: 70% / O: 30%]

Data

- Result based on quantitative analysis [A:40% / O: 10%]
- Available [A: 75% / O: 50%]

Figure 10 Reproducibility of papers published in Bioinformatics (N=100). (Hothorn & Leisch, 2011)

We note that for Original (O) papers the code availability is similar to what Vandewalle et al. (2009) found in papers in another domain (signal processing) but that in general sharing practices of code and data appears to be higher in *Bioinformatics* than in another domain journal like signal processing (IEEE Transactions on Image Processing). That said, a larger scale study with a set of common criteria may provide further evidence for that statement. Besides, the papers in *Bioinformatics* were collected at the beginning of the year 2010 and the study on signal processing used papers from 2004 and the effect on data sharing of this time difference is unknown in the present case.

Such studies were also conducted in Software engineering and criteria that makes a study in this field reproducible have been suggested (González-Barahona & Robles, 2011; Menzies & Shepperd, 2012). They define reproducibility as “the ability of a study to be reproduced, in whole or in part, by an independent research team” (González-Barahona & Robles, 2011, p. 77). Interestingly, they also base their reproducibility study on a Knowledge discovery process (Fayyad et al., 1996). They add one step at the beginning wich is “data retrieval from repository” but drop the “interpretation/evaluation” step as they state it to be not relevant for reproducibility.

3.1.3 Replication

More linked to *repeatability*, Menzies & Shepperd (2012) investigate issues of conclusion instability and the (statistical) factors leading to that state. This paper gives some insight on what happens internally (in the methods) applied. The importance of the methods is under covered in Reproducible Research papers. We may argue that it is quite logical as reproducible research does not focus on whether the conducted research yields “true” conclusions but is simply assessing if we have enough information to generate similar results.

In other words, the impacts of methods or sampling, experimental conditions etc. might create a state of *irreplicability.* Certainly if we consider computational experiments as dependent on *wet lab* interventions that may be expensive, time-consuming or dependent on precise conditions. Or what about biological events found in one single patient?

3.1.4 The answer of publishers

While some authors (Ioannidis, 2014) argue that the entire publication culture should be reviewed to enable RR, publishers react with policies or even new applications to link and visualize data in online papers or are partnering with initiatives to achieve this, like Elsevier and https://www.mysciencework.com. In other words, a sense of interactivity is embedded with online papers (illustrated by iPLOTS by Elsevier[3], despite its simplicity and the lack of support for complex data sets).

So, publishers are transforming their publication models and add widgets to online papers for visualization and presentations (audio files). The impact on the success or failure of new "publication" tools sometimes advocated (like research objects) is unknown. Particularly, to what extent do these initiatives annihilate any alternative way of publishing knowledge should also be investigated to judge the worthiness of efforts to replace or supplement academic knowledge dissemination with ROs, for instance.

3.1.5 Virtual experiments

From a user's perspective, especially while not really interested in the code running in the pipeline, another presentation that acts as the counterpart of lab experiments is an aggregation of models and protocols in a virtual experiment (Cooper, Science, & Road, 2014). These virtual experiments also address the burdens of replication but on a more statistical level where a user would be allowed to reuse models that would be suitable to the experiments protocols at hand. Mainly by separating these models, as found on https://www.ebi.ac.uk/biomodels-main/, from protocols to enable what we refer to as "scenario analysis" in the last chapters.

We might see virtual experiments as the computational model complement for code, data and executables sharing and reuse advocated by RR. Eventually, both virtual experiments and computational material availability and execution aim at improving reproducibility from "full replication" to conceptual replication, i.e. changing some aspects of an experiment (subcomponents). For this work and the prototype, no such models as available on BioModels in system biology were used but only packages for normalization, which are actually implementing statistical models but may be less elaborated. Still, we can prepare the functional architecture of our system to welcome these new initiatives and ask people from the field how they believe virtual experiments could satisfy their requirements for data analytics. Implementation and annotation of models and protocols are out of the scope of this work.

3.2 Some threats to share

3.2.1 License

In short, what's data is not evident to define, legally (Stodden, 2009). Even more when it comes to licenses, copyrights or privacy. Here, we escape the issue by placing it "out of the scope" as our research was done on available data (internal or external) and that the position of "sharing everything" is also not really defended. Nonetheless, this legal perspective on data sharing is of course an important obligation that is part of a data management plan.

3.2.2 Reproducible Research

Reproducible Research was discussed previously and its relation with *verification* as a "must-do-it" in science might actually not positively serve the need for data sharing. At least, according to Borgman

[3] https://www.elsevier.com/books-and-journals/content-innovation/iplots

(2012). Among the four incentives to share data (verify, serve public interest, ask new question and advance research) RR is judged as the most inadequate one. What the author points is the relativeness of the notion of verification or replication which is loosely defined by RR proponents. It is too domain specific or time bounded for some measurements. Moreover, tools may not be available (e.g. proprietary). The issue raised is valid, but isn't it closer to what was called *replicability* by Drummond (2009)?

> *"From an epistemological perspective, reproducibility and verification are the most problematic of the four arguments for sharing data. Often the research creativity lies in identifying a new method required to approach an old problem. Research outcomes often depend much more on interpretation than on the data per se. Separating data from context is a risky matter that must be balanced carefully against demands for reproducibility."* (Borgman, 2012, p. 1069)

3.3 Solutions from IT

3.3.1 Literate Programming and authoring tools

Assuming that a computational biologist is ready to share some data or simply to retrieve an older analysis, the question is what should be retrieved? Data and code in raw formats or something else? This would be identical to sending a zip file per mail with csv formatted files and scripts… So, how could this retrieval be potentially made more convenient? This part of the story starts in the eighties with the concept of Literate Programming invented by the creator of TeX and author of the *Art of programming*, Donald E. Knuth (1984).

The concept of *Literate programming* depicted here is cited in a large number of papers addressing reproducible research. Mostly describing tools organizing code and explanations for human readers into a single or multiple files (Hoefling & Rossini, 2014; Liu & Pounds, 2014). Originally, Donald Knuth presented *Literate programming* as a meta-language (WEB[4]) stored in a single file and capable of generating human readable information (by a process called *weaving*) in TeX and machine code (*tangling*) in Pascal (Knuth, 1984). Although this technique has been challenged by larger pieces of software that do not fit in a single file as we will see next, these are two transformations targeting both humans and machines. To illustrate the usage of *Literate Programming* and its application in reproducible research, a brief overviews of tools is provided (see Table 3). Then, the concept of compendium (R. Gentleman & Lang, 2007) is introduced and how it deals with the problem of reproducibility.

NAME	REFERENCE	DESCRIPTION	LANGUAGE
SWEAVE	Leisch (2002)	Implements literate programming concept to produce documents using LaTEX	R
KNITR	Xie (2014)	Similar to Sweave but with more outputs.	R
R MARKDOWN	Baumer, Cetinkaya-Rundel, & Bray (2014)	Based on Knitr. Good integration with RStudio and publication of dynamic documents on Rpublish. Authoring is done with the very light Markdown syntax. Generates html, PDF or docx (Word)	R

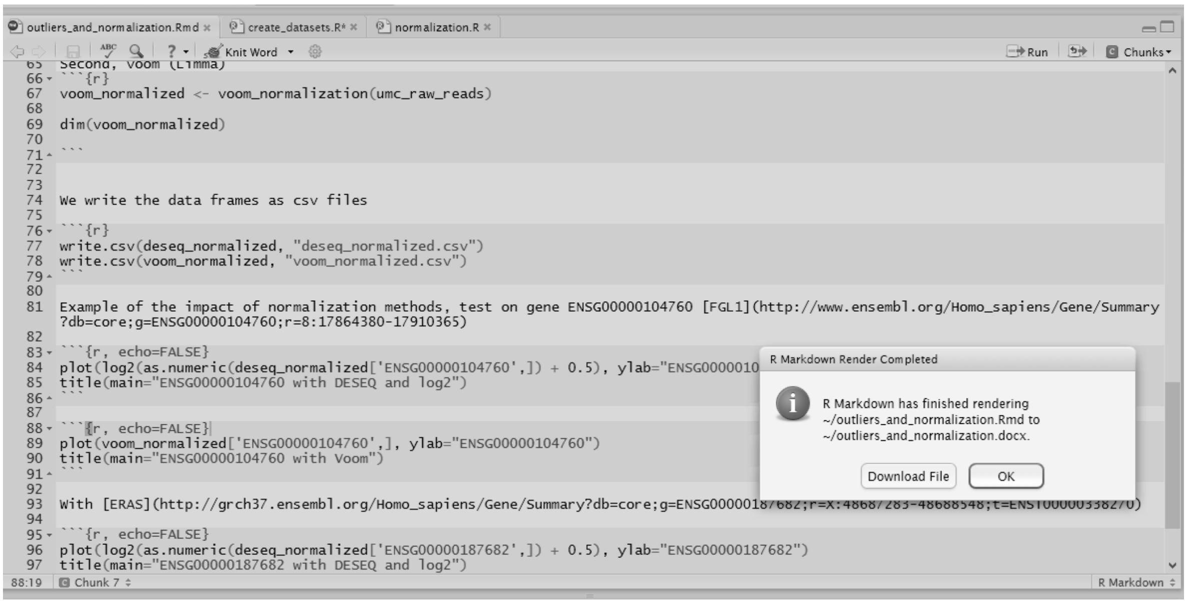
[4] Unrelated to the World Wide Web that we are using every day. As Knuth explains and for the anecdote: "I chose the name WEB partly because it was one of the few three-letter words that hadn't already be applied to computers." (Knuth, 1984, p. 97) .

IPYTHON NOTEBOOK	Pérez & Granger (2007)	Extended by the Jupyter project, a notebook is also a dynamic document. Notebooks can integrate dynamic plots made by matplotlib or even bokeh (a python module that was also used to build the user interface). They are designed according to the same principles than RMarkdown.	Python

Table 3 Authoring tools

First, from a technical perspective, Sweave is an R authoring package that implements the most genuine concepts of *Literate Programming* as developed by Knuth but for the R language instead of Pascal. Then, based on the same principles, Knitr takes some liberties in the output formats (HTML, Markdown,…) and offers therefore different representations of the documented code written in R. RMarkdown (based on Knitr) defends an easier approach to generate these dynamic documents (with text, code, graphs…) by allowing authoring in Markdown and offering a better integration with RStudio. These tools should not be confused with code documentation (Javadoc, Sphinx…) generators although they actually document code. To create such a document, the code (i.e. R code) is executed and plots are generated and transformed into other formats. For instance the code that creates a plot will be executed and the plot inserted in an HTML document as a static image for instance. So both the code and the explanations are in the same web page or PDF file. The output is selected in advance by the author of the dynamic document. In Figure 11, we generated (or knit) a word document from a Markdown file in R studio. The greyer areas are code that will be transformed and the white areas contains text that will be formatted in the output document. A link tag will become clickable in the final document, for instance.

IPython notebook is also a dynamic document generator, like RMarkdown, but for the python environment instead of R. The Jupyter project is a new extension and foundation of the IPython notebook which also adds additional programming languages (like R) but is still in an early development stage at the time of writing.



~~~
Second, voom (Limma)
```{r}
voom_normalized <- voom_normalization(umc_raw_reads)

dim(voom_normalized)

~~~

We write the data frames as csv files

```{r}
write.csv(deseq_normalized, "deseq_normalized.csv")
write.csv(voom_normalized, "voom_normalized.csv")
```

Example of the impact of normalization methods, test on gene ENSG00000104760 [FGL1](http://www.ensembl.org/Homo_sapiens/Gene/Summary
?db=core;g=ENSG00000104760;r=8:17864380-17910365)

```{r, echo=FALSE}
plot(log2(as.numeric(deseq_normalized['ENSG00000104760',]) + 0.5), ylab="ENSG0000010
title(main="ENSG00000104760 with DESEQ and log2")
```

```{r, echo=FALSE}
plot(voom_normalized['ENSG00000104760',], ylab="ENSG00000104760")
title(main="ENSG00000104760 with Voom")
```

With [ERAS](http://grch37.ensembl.org/Homo_sapiens/Gene/Summary?db=core;g=ENSG00000187682;r=X:48687283-48688548;t=ENST00000338270)

```{r, echo=FALSE}
plot(log2(as.numeric(deseq_normalized['ENSG00000187682',]) + 0.5), ylab="ENSG00000187682")
title(main="ENSG00000187682 with DESEQ and log2")
```

R Markdown Render Completed

R Markdown has finished rendering
~/outliers_and_normalization.Rmd to
~/outliers_and_normalization.docx.

Figure 11 Screenshot of the edition of an RMarkdown document in RStudio (authoring tool)

3.3.2 Compendiums and Hypermedia

Gentleman & Lang (2007) extended the concept of Literate Programming (Knuth, 1984) and proposed the concept of Compendium as an aggregation of dynamic documents that are *transformable*. So, the elements of the dynamic document can be transformed into different *views* (i.e. how to communicate the code to users). In their paper *Statistical analysis and reproducible research* they implemented a prototype using R and Sweave. Here the goal is to structure context and artefacts in *executable* "units" of software. These units (the compendiums) are exchangeable and enable independent review of the computations made.

The authors state the 5 following goals of compendiums (R. Gentleman & Lang, 2007):

1. **Encapsulation**: the work of an author can be inspected by diving into the original documents (that may have various formats)
2. **Easy to re-run** also with alternative inputs
3. **Adaptable** and allow method extension: enough details are present in a compendium for this.
4. **Programmatic document construction**: equivalent to RMarkdown/IPython, which are therefore not strictly speaking compendium *generators* but *transformers* to create a view (human readable document).
5. **Manipulation of documents** in many fashions. Might be including specific views targeted to particular audiences (independent research, to students for a course (less details)…)

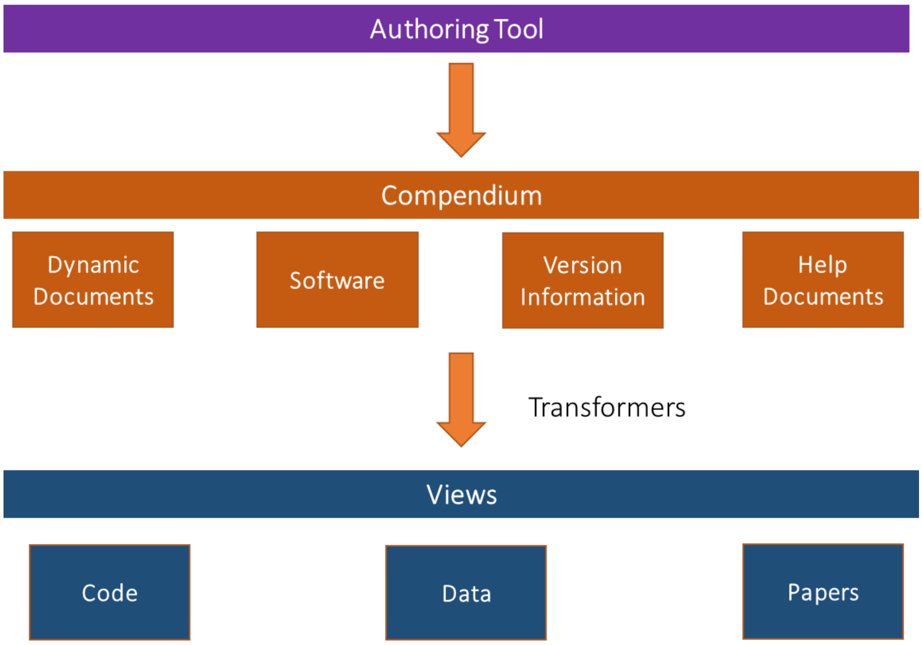
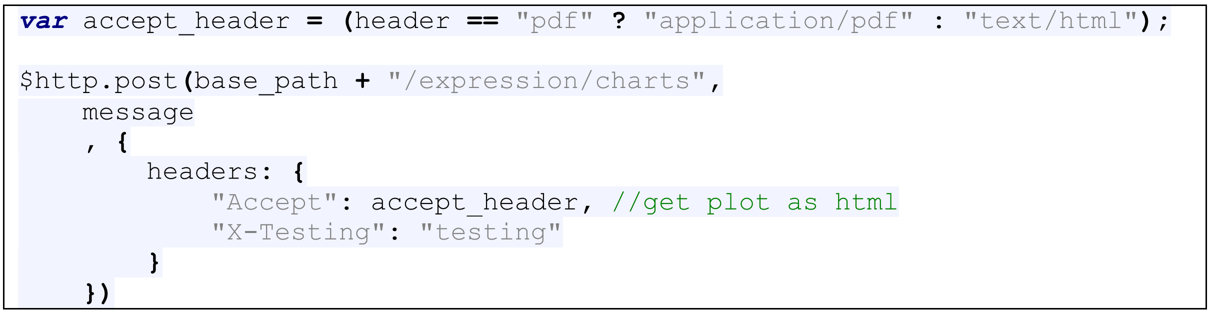


Figure 12 Compendiums and their components

But except from Sweave, very little technical insight is provided on how to communicate the content of a compendium with "transformable" components. This is where the design choice based on web resources is made. By treating compendium elements as web resources we attempt do leverage existing mechanisms to build an "open" compendium. In other words, strong relations can be made between compendiums (as an aggregation of dynamic documents) and *linked data* using the HTTP protocol. More fundamentally, two aspects of this protocol: (1) identifiers (aka Uniform Resource Identifier – URI – described in RFC3986) and (2) the "Accept" header – described in RFC7231 - in an HTTP message which can call for any *representation* of a resource recognized by a server (e.g. the same information as JSON or as an HTML page).

A web resource can have one or more *representations* which is potentially a dynamic document, but could also perfectly be communicated as a source code, a downloadable executable or package just by setting the *Accept* header appropriately.

Next, we imagine a scenario where we c*reate* the same plot, for one data set and one gene. The first time as a dynamic plot and the other time as a PDF export.



```
var accept_header = (header == "pdf" ? "application/pdf" : "text/html");

$http.post(base_path + "/expression/charts",
    message
    , {
        headers: {
            "Accept": accept_header, //get plot as html
            "X-Testing": "testing"
        }
    })
```

Figure 13 Code fragment showing how to "query" an interactive plot or a URL to a pdf

In case the accept header is set to "text/html", an interactive plot will be generated. If set to "application/pdf" a URL to a PDF file is generated. This simple example shows how we get back to this notion of transformation that compendiums put forward but this time for any web resource. Generalization to any kind of resources is perfectly allowed by the "web architecture". In the RNASeqTool the main resources were methods, packages, charts, data sets and virtual experiments. These elements are described in Chapter 6.

Setting more precise constraints on resources (which representations they support) might eventually lead to an API behaving as a compendium and serving multiple representations of a resource.

3.3.3 Research Objects

Research objects can be viewed as an aggregation of tools or pieces of software mixing code versioning with sharing short statements, e.g. Nano-publications (Mons & Velterop, 2009). Technologies supported by ROs are listed on http://www.researchobject.org/. A second view which is put forward in papers are workflow-centric research objects falling under the Workflow4Ever project, a group of academic initiatives to promote workflows and research objects (http://www.wf4ever-project.org).

They are built upon the notion of resource (but not explicitly *representations)*. The missing link between Linked data and Research context are presented in the form of Research Objects (RO). The research objects principles (listed in Table 4) might appear familiar to readers dealing with data management. Indeed, there is a strong overlap between the two concepts and there is every reason to believe they should be combined.

Principle	Explanation
Reusable	A RO is a "package" that can be reused in another experiment (totally or partially)
Repurposeable	A package aggregates methods, data, processes etc. that should be accessible individually (partial reuse) for other analyses
Repeatable	Enough information for intern or external understanding of the experiment.
Reproducible	Provide same inputs for validation of the study by external researchers
Replayable	Ability to redo the "workflow" but not necessarily by executing it but rather investigating provenance and results metadata
Referenceable	Should include an authorship mechanisms (cite a Research Object)

Revealable	Same as “repeatable “ but for auditing purposes
Respectful	Deals with the issues of replacing papers by ROs, intellectual property, reward mechanisms etc.

Table 4 8 Principles of ROs (Belhajjame et al., 2014).

Hence, a small introduction to the FAIR principles is provided. Next we deal with a very short overview of research objects and their main ontology OAI-ORE which has been extended to a RO ontology in the literature.

3.3.4 FAIR

The fair principles are a set of internationally discussed guidelines for FAIR data management and engage semantically enriched data together with persistent identifiers (PID) for document retrieval. The main criteria of FAIR objects are:

- Findable: objects are denoted by a unique identifier which does not change over time and the usage of metadata.
- Accessible: Appropriate authorization management
- Interoperable: Human and machine readable
- Reusable: data objects can be quoted and linked to other data sources

FAIR principles are applied to data objects (i.e. combination of PID, metadata and *data elements*[5]). A complement to these data objects would be research objects (RO), they are discussed in the next section.

3.3.5 Research Context

Besides FAIR data objects (DO), research context (hypotheses, assumptions, goals, methods) might be seen as FAIR research objects (RO). Researching what aspects of metadata and data elements that makes a data object FAIR might lead to a more general notion applying these principles to the research context. In the latter case, FAIR research context is designated by the terms Research Object (RO). To achieve that, it is assumed that presenting a user-friendly interface (dashboard like) to members of a research staff for both traceability of data and context might leverage the usage of DOs and ROs and subsequently the reproducibility of computational experiments.

It is to note that this strong overlap between DOs and ROs is also highlighted by the Force11 community itself[6] which subscribe to the “8-R”of Research objects (Table 4). The ultimate goal is that this combination eventually works for end-users collaborating inside an organization or between different organizations, research groups…

Next, we attempt to map our architecture and feature choices to an ontology that is commonly applied or extended in research objects (Bechhofer, De Roure, Gamble, Goble, & Buchan, 2010). We did not implement these messages in our technical artifact so this mapping is fairly theoretical. We start with a small table (Table 5) showing that the choice of the web architecture in both settings (RNASeqTool) and OAI-ORE (which is an ontology that is of course also embedded in the web architecture and RDF). That is, the dynamic and exploratory concepts that biologists manipulate to the raw machine readable objects that could be exchanged.

[5] https://www.force11.org/node/6062/#Annex1 , also http://www.dtls.nl/fair-data/

[6] https://www.force11.org/node/6062/#Annex4

CONCEPT RNASEQTOOL	OAI-ORE	DESCRIPTION
VIRTUAL EXPERIMENT	Resource map	Describe collection of resources
REPRESENTATION	Representation	See HTTP
RESOURCE	AggregatedResource	Object of interest (data set, method…)
PUBLIC IDENTIFIER	URI	Unique path to an object

Table 5 Mapping RNASeqTool end-user and OAI-ORE

In short, ontologies adopted by research objects closely maps to fundamental aspects of the web architecture and how this prototype was designed. This makes one of the representations served as enriched RO (annotated data about an experiment) quite straightforward. This area needs to be investigated further, certainly because it shows that workflow-centric ROs are not sufficient nor ideal in our situation.

To finally link Research objects to our Web API we show an example based on a RO as build by Belhajjame et al. (2014). We see that there are also presented as an aggregation of resources and that we might eventually plug-in resources created by the tool in the description of a research object instead of a file This is illustrated in Figure 14 where the original RO shown in Belhajjame et al. (2014) is adapted with resources generated by the RNASeqTool.

```
<> a ro:ResearchObject;
dct:title "Outlier analysis";
dct:creator </foaf/wigard>;
ore:aggregate < DESeq_1_18_0_umc_read_counts_table_without_8433.csv>, <
exp_umc_outlier>,<kmeans_outliers_DESeq_1_18_0_umc_read_counts_table_with
out_8433_74304198-3f7a-4b7c-a83d-83c3ebf7558e>.

< DESeq_1_18_0_umc_read_counts_table_without_8433.csv> a ro:Resource.
<exp_umc_outlier> a roterms:Hypothesis.
<kmeans_outliers_DESeq_1_18_0_umc_read_counts_table_without_8433_74304198
-3f7a-4b7c-a83d-83c3ebf7558e> a roterms:ProspectiveRun.

dct:created [...]
```

Figure 14 A tentative research object described by an ontology based on Belhajjame et al. (2014)

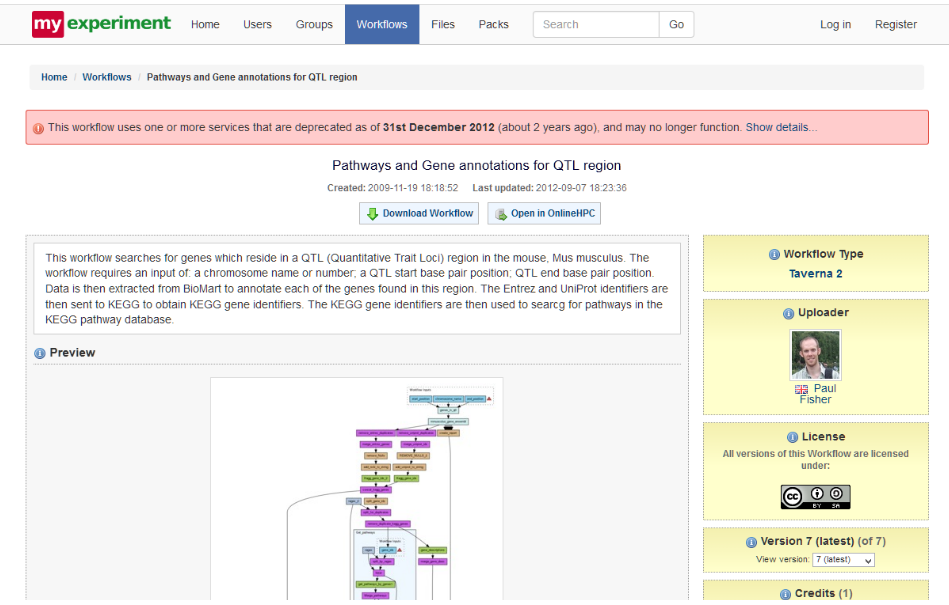
With these explanations, we offer some intuition on how to combine the power of compendiums and the description of a virtual experiment as ongoing research on ROs attempts to provide. As such, ROs would bring the machine interpretable complement to virtual experiments that are targeted to human users.

3.3.6 Link with workflows

NAME	REFERENCE	EXPLANATION
VISTRAIL	Callahan et al., (2006), Freire et al. (2014)	Workflow generation from exploratory analysis and tracking parameters used and charts generated. Integration with python script via PythonSource.
TAVERNA	Oinn et al. (2004)	Serialize workflows and the role of web services. Acts as an orchestrator.
GALAXY	(Goecks et al., 2010)	Provide a graphical interface to command line tools. Serializes workflows and make data retrieval or processing easier with its connections to third-parties like biomart.

Figure 15 Reproducible workflows

Research objects and compendiums suggest to gather information about the context (like the hypotheses) together with workflows (if any). Once more, Taverna workflows and research objects are not incompatible. A special type of research objects is even fully focused on the integration of context and workflows.



This workflow uses one or more services that are deprecated as of **31st December 2012** (about 2 years ago), and may no longer function. Show details...

Pathways and Gene annotations for QTL region

Created: 2009-11-19 18:18:52 **Last updated:** 2012-09-07 18:23:36

This workflow searches for genes which reside in a QTL (Quantitative Trait Loci) region in the mouse, Mus musculus. The workflow requires an input of: a chromosome name or number; a QTL start base pair position; QTL end base pair position. Data is then extracted from BioMart to annotate each of the genes found in this region. The Entrez and UniProt identifiers are then sent to KEGG to obtain KEGG gene identifiers. The KEGG gene identifiers are then used to searcg for pathways in the KEGG pathway database.

Workflow Type
Taverna 2

Uploader

License
All versions of this Workflow are licensed under:

Version 7 (latest) (of 7)

Credits (1)

Figure 16 Illustration of myExperiment workflow items.

As can be seen from the alert (red message) in Figure 16, keeping workflows executable after some time is an issue (known as workflow decay). The trend of web services or packages to decline is still technically unsolved in this thesis but several technologies that may provide some stability between the interacting components of a virtual experiment have been suggested like the *executable paper*. It was proposed in the "Executable Paper Grand Challenge"[7] organized by Elsevier in 2011. One of the recommended solutions is described in *SHARE: a web portal for creating and sharing executable research papers* (Van Gorp & Mazanek, 2011). This solution, in short, puts remote access to virtual machines (with the required software installed) forward. A link from a reference in a paper allows a reader to access a virtual machine and consult the tools and data used for the given research. First, Executable papers are providing a *link* to computational resources, here in the form of a virtual machine. Second, an executable paper editor was built by the authors to help researchers design their virtual machine and create a link to it.

CONCEPT	ARCHITECTURE	HUMAN-MACHINE	IMPLEMENTED
LITERATE PROGRAMMING	File	Transformation	Dynamic documents
COMPENDIUM	File or web service	Transformation	No
RESEARCH OBJECT	Web	Ontologies	myExperiment.org ROHUB.org

Table 6 State of the most important technological solutions found

As will be seen in the next chapters, the workflow-centric approach usually implemented by either research objects or Taverna does not fit all the requirements of exploration and scenario analysis. These activities derive from a workflow but are more trial-and-error/exploratory than procedural.

[7] http://www.executablepapers.com/

Chapter 4 Data analysis and visualization of RNA-Seq data

Here we provide the basic elements to understand what is going on in the fields of data analytics for RNA-Seq data. An introductory but more explanatory description of the pipeline is provided in the chapter about knowledge discovery. Here we address the techniques of differential expression analysis, outlier mining ("novelty detection"). But we start first with what differential gene expression means (both biologically and in terms of counts).

4.1 Theoretical background

4.1.1 Interactivity

As we recall from the problem statement, interactivity is the second aspect of this work after reproducibility. The thesis started on a dual problem: more interactive tools are needed and at the same time "reproducibility" for experiments involving computational material is presented as dramatically deficient. We have seen that Reproducible Research itself lack consistency or coherent measurements of what makes an experiment replicable. For this part, interactive data analytics, we present some standards and tools for RNA-Seq data as implemented in the prototype.

Holzinger & Jurisica (2014) call for a merge between two disciplines to tackle the constant need of large data sets analytics: Human-computer interaction and knowledge discovery from data.

> *"The idea of the HCI-KDD approach is in combining the "best of two worlds": Human‑Computer Interaction (HCI), with emphasis on perception, cognition, interaction, reasoning, decision making, human learning and human intelligence, and Knowledge Discovery & Data Mining (KDD), dealing with data-preprocessing, computational statistics, machine learning and artificial intelligence." (Holzinger & Jurisica, 2014, p. 6)*

Next, Holzinger, Dehmer, & Jurisica (2014) suggest four areas where interactivity should be increased. These areas are covering the KDD process (Fayyad et al., 1996) but adapted to Life sciences. In short, the authors suggest to make everything interactive (merging & integration, pre-processing, mining and exploitation). That is basically what the design of our prototype followed, implementing buttons for all these steps. They note a cross-disciplinary aspect of Knowledge discovery in a biomedical informatics context but are vague on the actors involved (bioinformaticians and – computational - biologists) and claim that data producers and users are a single entity (Holzinger et al., 2014). Figure 17 illustrates the activities which an end-user interactively performs during a KD process according to Holzinger et al. (2014).

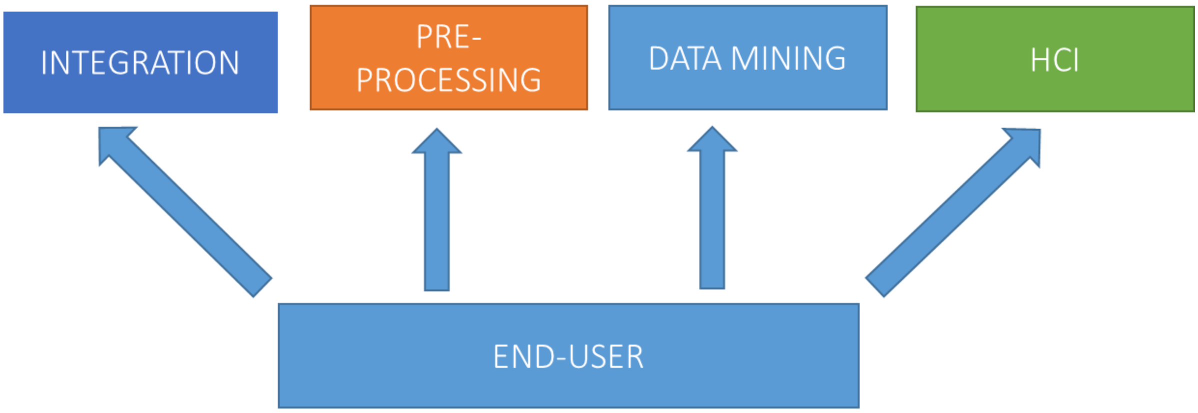


Figure 17 KDD in Life sciences, activities adapted from Holzinger et al. (2014)

4.2 Data analytics for RNA-Seq data

Chapter 5 provides more technical details about the RNA-Seq read counts file that was processed by the prototype. For this section, we take for granted that we start with a file that is nothing more than a matrix with samples in columns and genes in rows. The content comprises counts which give an indication on the “activity” of a gene. Gene expression is simply put the level at which a gene is active. Or in biological terms: the level at which a gene is transcribed into RNA.

4.2.1 Bioconductor

Many packages manipulated during data analysis for sequencing data are managed by *Bioconductor* which is an R package repository focused on biological computations (R. C. Gentleman et al., 2004). Reproducibility is one of the goals of *Bioconductor* (R. C. Gentleman et al., 2004).

Studies providing extra information about the methods used may do it with an R package as it the case for De Sousa E Melo et al. (2013). In that precise situation, data sets and the pipeline to *reproduce* the analysis is provided in one single package (see “DeSousa2013”[8]).

4.2.2 Data exchange MIAME and MINIM (ROs)

Initiated by the functional genomics data society (FGED) the design choice here is to agree on a minimal sufficient information description of a micro-array or RNA-Seq (an extension of the micro-array format) experiment. This standard is also compatible with some Bioconductor R objects is a well-known standard for micro-array or RNA-Seq data exchange. It shows how some existing metadata standards conveniently support a “sufficient” information strategy. We also note that sufficient is not a stable attribute and that standards are evolving, as illustrated by the challenges of “minimality” of the MIAME standard (Brazma, 2009). The interested reader is redirected to https://www.biosharing.org where standards/ontologies covering a wide range of biological experiments are referenced.

REPOSITORY	DESCRIPTION	URL
ARRAYEXPRESS	Access expression profiles	http://www.ebi.ac.uk/arrayexpress/
GENE EXPRESSION OMNIBUS (GEO)	Data reuse of previous experiments	http://www.ncbi.nlm.nih.gov/geo/

Table 7 Two widely used data repositories that are MIAME compliant

The MIAME is an interesting metadata standard to look at because it reveals both some contextual information and the data that are expected to help further reuse. MIAME is supported by plain-text files (SOFT) or XML (MINiML). Using an example of MIAME compliant file from GEO, AMC colon cancer AJCCII[9] (De Sousa E Melo et al., 2013; Kemper et al., 2012) we can easily see that this standard presents information about:

- Experiment (description)
- Samples
- Platform
- Authors
- Relation to other projects

[8] http://www.bioconductor.org/packages/release/data/experiment/html/DeSousa2013.html
[9] http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE33113

- Evolution (submission and update dates MM/DD/YYYY)

This goes with different levels of granularity. It means that the standard supports information from related projects[10] to indications per sample[11] and raw data. Based on an identical way of thinking, another RO model was suggested based on MIM standards. This model is for workflow-centric ROs and is called MINIm (Zhao et al., 2012).

4.3 Solutions from IT

Numerous tools, languages, frameworks or cloud solutions emerge to tackle data in Life sciences. Such richness is impossible to reduce to a single chapter. Hence, a micro-overview of some libraries or packages that were implemented in our prototype (see Chapter 6) are discussed.

4.3.1 Data analysis with R

As has been explained previously, R is a big player in the field of data analytics and scripts and is well known in life sciences with Bioconductor. In our quest to execute third party tools or packages, a binding to R was mandatory. Most of the methods for normalization of RNA-Seq counts are made available through R packages on Bioconductor and the prototype is mainly built with python. Here we describe four of these normalization or transformation methods and packages implemented.

These methods are applied without considering replicates and uses the normalization procedures to make samples comparable. A replicate would be cells from a patient before and after treatment, for instance. Here, all samples are considered as independent from each other. These methods all originally work for differential expression analysis and with replicates. Some manuals of packages like DESeq2 (Love, Huber, & Anders, 2014) emphasizes that working without replicates is possible but results should be taken cautiously. Once more, we will not dive into great details about the packages but just justify their presence in the prototype and how well they illustrate the "model jungle" that we are also addressing in Chapter 5.

NAME	REFERENCE	PURPOSE	SHORT DESCRIPTION
DESEQ	Anders & Huber (2010)	Outliers	The median of the division of the values in a column (sample) by its *geometric mean* gives a size factor (which will increase or decrease the raw count).
DESEQ2	Love, Huber, & Anders (2014)	MDS/PCA/Clustering	Here we applied the RLOG transformation, so DESeq2 helps us to cluster samples or reduce dimensions.
EDGER	Robinson, McCarthy, & Smyth (2010)	Outliers	By default, the calculation of a size factor is done with the trimmed mean of M-values normalization method (TMM). Here the authors were concerned about the assumption that other methods use a proportion of counts for one gene divided by the total count of the library (i.e. a sample). However different conditions yield different proportions which cannot be exactly compared (M D Robinson & Oshlack, 2010).

[10] http://www.ncbi.nlm.nih.gov/bioproject/PRJNA156585

[11] http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM820048

LIMMA/VOOM	Law, Chen, Shi, & Smyth (2014) Smyth (2005)	MDS/PCA/Clustering	Limma existed for continuous values generated by micro-array experiments. Voom, a function of this package, transforms discrete counts to log-counts per million and precision weights to make the whole package suitable for discrete data from RNA-Seq, after "vooming". It can be combined with TMM for instance to build a workflow normalizing and transforming the original counts (G. K. Smyth, Ritchie, Thorne, Wettenhall, & Shi, 2013).

Table 8 R Packages used for normalization

This offers an overview about the type of ongoing discussions and why there are different models that are strictly embedded in debated assumptions around the RNA-Seq technologies. Next, one example of an analysis that was not our outlier mining. We see that Limma has internal functions to visualize samples and get informative results A common analysis that we did not implement in the prototype is DE which is supported by all the packages shown in Table 8.

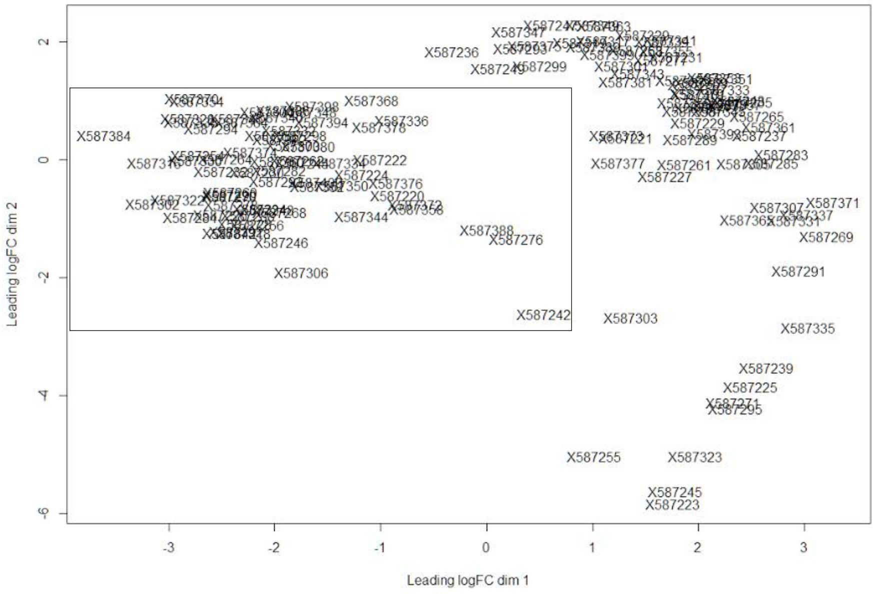


Figure 18 Limma MDS on transformed data

As an example of application of a package on data, Limma provides a method of dimensionality reduction in addition to differential gene expression. By correcting for noise in the data it is possible to see differences in the samples based on the counts present in the data set. Inside the square of Figure 18 are the tumor samples and outside are the control samples. An MDS type of analysis (unsupervised) on all genes (when log transformed[12]) is able to show a distinct difference between tumors and healthy tissues (positive controls – PC) in the data set.

[12] Data set is Limma_3_22_7_genentech_read_counts_table

4.3.2 Data analysis with Python

The basic data structure for the entire prototype is a *pandas dataframe*. Pandas is a statistical library in python which makes it easier to conduct descriptive analytics. More advanced data analysis can be performed with modules like scikit-learn and statsmodel. The last two modules are combined with pandas through the underlying *numpy* module.

4.3.3 Data visualization with Python

If we recall that compendiums constraints impose transformation of components to make an investigation easier for researchers, it will not be surprising that our plotting pipeline is just about resources and transformations. To achieve this, a standard plotting library in python *matplotlib* or *bokeh* were both utilized. While bokeh directly supports HTML rendering, matplotlib plots were converted with *mpld3*. The fundamental difference is that, at time of writing, bokeh utilizes the HTML canvas whereas *mpld3* generates SVG with d3.js. All in all, a reflection has to be observed when applying these techniques with plots generating a lot of data points. One preliminary recommendation would be to use *bokeh* and *bokeh server* that also implement streaming capabilities for large data sets.

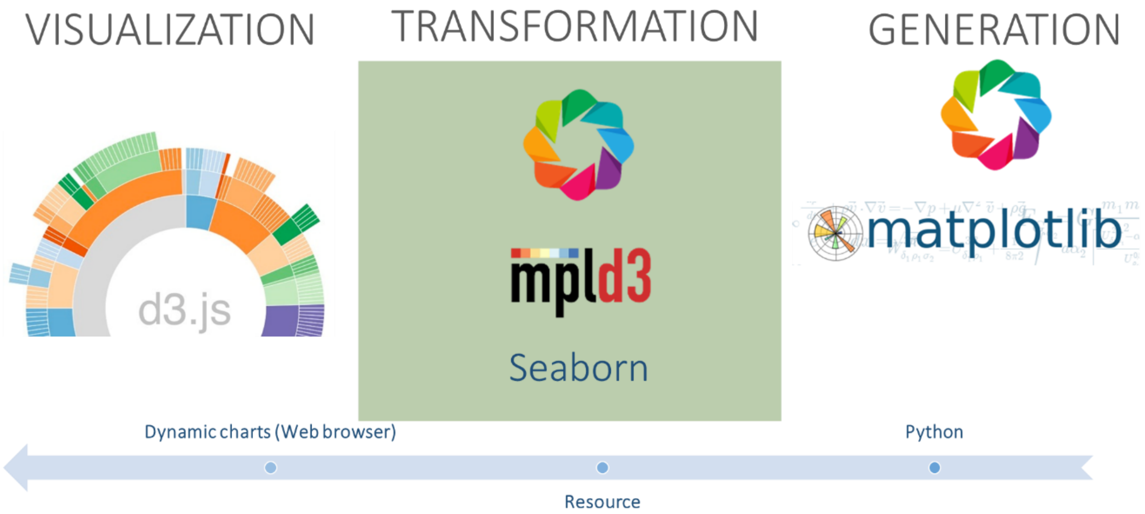


Figure 19 Creating dynamic plots for the web in python

Despite pre-build interactivity features, both technologies are still quite recent and their challenges are hard to tackle. Transforming complex charts or plots to render identically in a python GUI, notebook and in a browser is complex. Customizing interaction requires a great deal of programming specific code chunks for *bokeh* or *mpld3*.

From a user-side, interactivity is quite limited to basic operations (zooming, scrolling). Nonetheless, encouraging roadmaps and the open source nature of *bokeh* might indicate that these systems will be improved. Recent additions of widgets might improve the "HCI" part of the website. Table 9 shows the packages and their role (purpose) in the prototype.

NAME	ENVIRONMENT	STEP	PURPOSE
PANDAS	Python	All	Data frame and descriptive statistics
SCIKIT-LEARN	Python	Mining	Clustering for outliers
DESEQ	R	Pre-processing	Normalization
DESEQ2	R	Pre-processing	Transformation

EDGER	R	Pre-processing	Normalization
LIMMA	R	Pre-processing	Transformation
RPY2	Python	Pre-processing	Python/R binding
BOKEH	Python	Visualization	Dynamic charts
MATPLOTLIB	Python	Visualization	Static charts and PDF export
MPLD3	Python	Visualization	Dynamic charts
SEABORN	Python	Visualization	Esthetics for plots in python and advanced visualization

Table 9 Main packages and modules for data analytics

Chapter 5 KDD in practice: an application in RNA-Seq analysis

In this chapter we illustrate a generic 6 steps knowledge discovery process by showing how the data sets of colorectal cancer patients were analyzed for outlier detection. This leads to a more realistic overview of how the KDD process is applied in bioinformatics from design to visualization. In this chapter we discuss the analysis of two data sets that are identified as umc and Genentech.

5.1 Knowledge discovery

García, Luengo, & Herrera (2015) propose a good summary of the rather different approaches or definitions that fall under Knowledge discovery or data mining (DM) processes. We do not discuss details related to particular processes like KDD (Fayyad et al., 1996) or CRISP-DM (Shearer, 2000) as we are satisfied with a more generic abstraction. Hence, the steps aggregated by García et al. (2015, p. 2) are presented here with a short description and depicted in Figure 20:

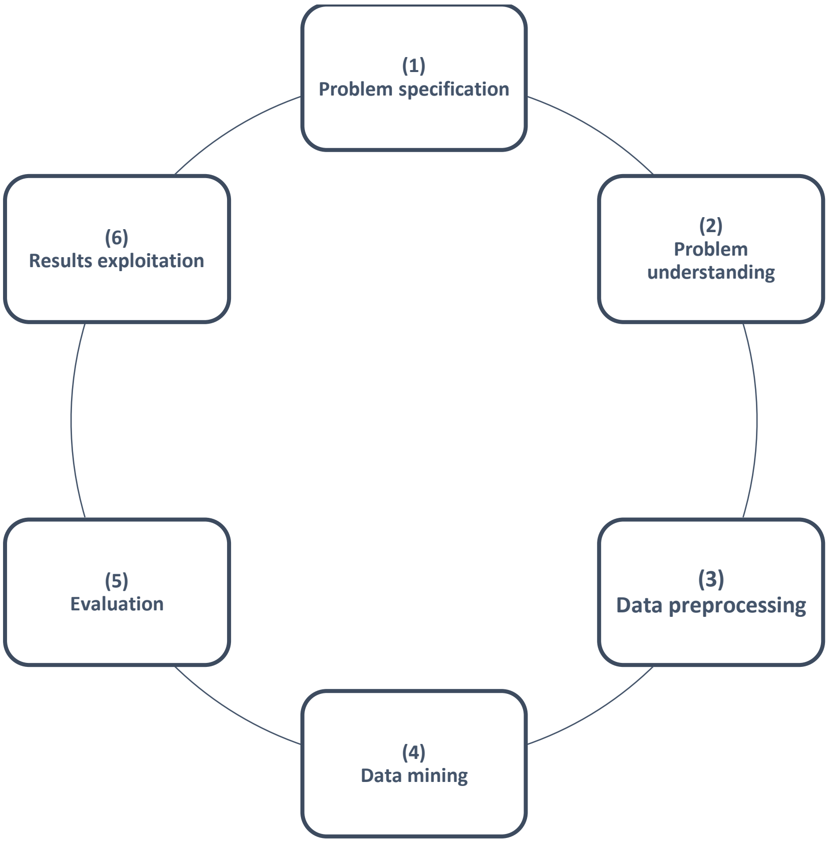


Figure 20 Representation of the main steps of a generic knowledge discovery process

1 - Problem specification: gathering of prior knowledge (e.g. expert interviews) and objectives of end-users

2 - Problem understanding: understand the data set(s) (e.g. by an exploratory data analysis – EDA (Tukey, 1977)) and links with experts' knowledge of their field to ensure a high reliability of data analyses and data products

3 - Data preprocessing: cleansing, noise removal, integration, transformation, reduction

4- Data mining: "Extraction of *useful* patterns from data" (Fayyad et al., 1996) or exception/anomaly/outlier detection (Hodge & Austin, 2004)

5 - Evaluation: models fit, (cross-)validation and interpretation

6 - Results exploitation: Visualization, exchange with third parties, interpretation by end-users

This list should not be interpreted like a straightforward or determined process starting with “problem specification” and having visualization as an output without any possibility to go backward. Moreover, apart from the data miner/statistician and end-user/expert, there are other actors that may be involved in this process, as it is the case in the KDD as originally presented by Fayyad et al. (1996). For instance, in bioinformatics and biomedical research collaboration and multidisciplinary approaches are illustrated by lab workers generating data based on blood or urine samples from patients and bioinformaticians/computational biologists manipulating algorithmic pipelines (thus creating “sub-KDD” processes where the results[13] serve as a basis for a new KDD cycle).

First, we will give the keys to understand the problem specification, which are roughly the research questions driving the biologists but also closely related to the particularities of the technologies involved (such as RNA-Seq). As research depends on these “wet”[14] side technologies and the protocols that are attached to it, the problem specification helps to retrace the story of how the data was generated together with the aspirations of the researchers.

In a second step, the data itself must be understood since any further results of analyses done on the data sets will depend on the pre-processing steps (step 3, e.g. normalization of RNA-Seq data). Some questions that one may ask are the following: Are there biological or technical replicates present, are metadata about patients available and, finally, how are the counts computed in the case at hand.

Third, the preprocessing step will be illustrated by “RNA-Seq data normalization”. Several techniques, assumptions and packages are briefly described. Mainly edgeR and DESeq which were introduced in section 4.2.1.

FEATURES/SAMPLES	S1	S2	S3	S4
F1	500	550	402	800
F2	0	0	160	1
F3	…	…	…	…

Table 10 Example of a fictional count data set

Fourth, this work does not especially contribute to the field of data mining per se but rather how it might be part of an interactive process initiated by biologists (user-friendly interface) in collaboration with bio-informaticians (code, APIs). Nevertheless, some basic features for outlier detection were implemented and explained in the section dedicated to this step.

Fifth, as it is the case of the data mining step, model evaluation does not exceed some evidence collected on the implications of different normalization methods applied on data. No new normalization method is proposed here and no new groundbreaking models for RNA-Seq data analysis are suggested. Step 4 and 5 are intensively debated by statisticians, computer scientists and mathematicians and we will refer the interested reader to related literature throughout this section.

Finally, results exploitation is depicted from the end-users’ side. The main goal here is to explore state of the art technologies to render visual elements in a browser while being able to control their

[13] intermediate data sets or secondary data usage

[14] Laboratory side

generation "server side" and keep track of relevant information for later re-use. Only technologies or libraries that are "compatible" with the web resources were investigated.

In the next sections we illustrate the different steps in the context of RNA-Seq data analysis, more precisely what the problems of the end-users are and what these counts are representing. These subsections apply the generic KDD steps without discussing reproducibility problems or implementations and are therefore recommended to get an overview of gene expression analysis based on RNA-Seq data from a more "biological" perspective.

5.2 Problem specification

At the beginning of this research project, discussions were held on a regular basis. This step is overlapping with requirements engineering which asks "how to find out what users *really* need?" (Goguen & Linde, 1993, p. 152). This activity dealt with loosely defined requirements, what has to be implemented and how and what kind of analyses techniques should be present. Requirements related to the "ideal" visualization tool are listed in Table 11. Only a subset of these requirements were actually implemented in the prototype.

ID	REQUIREMENT	EXPLANATION
1	See reads coverage per gene, exon, nucleotide	For this, different data sets are needed and graphical capabilities close to IGV should have been integrated.
2	Visualization of reads coverage with IGV (external viewer, Integrative Genomics Viewer)	This could have been resolved with a link from the files (BAM files) directly served by the API. We note that BAM files are way bigger and resource demanding to process
3	Link with clinical data: a patient file covering a subset of all the samples present in the datafiles (Age, recession, stage of tumor).	An incomplete file with data from patients was available but the way to integrate them with the application (data quality, important attributes) would have needed a separate project
4	Sequencing platform for a sample (HiSeq/NextSeq)	Again a step that would benefit from automatic processing. This information is also available in BAM files.
5	Links to Ensembl (adapted as a copy of gene metadata imported from ensembl's biomart)	Ensembl is one of the "reference genome" providers. Reference genomes also have their version, here it's not the las version that is used but the GRCh37 release 78. Genome versions are an extra big challenge to integrate with interactive tools.
6	Factor analysis, PCA analysis or clustering of samples based on all or subset of genes.	Here on a table of counts, samples or genes must be visualized with different techniques. These techniques were clearly suggested without any confidence that they were appropriate techniques for the type of data set at hand. This is discussed in the next section "problem understanding".
7	Generate a heat map of all genes for all samples	Computationally intensive around 40000 exploitable entries (removing genes with zero values everywhere).
8	Ability to conduct analyses similar to previous studies (clustering and prediction)	A study caught the attention because the authors designed a classifier (De Sousa E Melo et al., 2013) for three cancer subtypes that are expected to be present in the sample.
	Fast and Apple® style design	The visualization interface has to be responsive, plots must be fast to appear on the screen and a catchy design is important.

Table 11 Requirements before starting

From this shortlist summarized from several weekly[15] meetings of approximately half an hour (4 meetings in total) hold in the first weeks of this project from December to mid-February, illustrating the rather explorative nature of this study, we can see that:

- One requirement clearly described a scenario of “reusable” analyses based on previous papers.
- Linked data to external tools is recurrent (Genome viewers – IGV - or databases)
- Linked data to meta data (clinical, technical protocols, sequencing platforms)

Initially, the product artifact (RNA-Seq mining tool) was expected to cover three types of analyses:

- Gene expression analysis: Compare genes, samples and data sets based on a table of sequence reads counts that represents gene expression.
- Fusion genes identification: Identification of somatic fusion genes based on chimeric junctions identified in paired-end RNA-sequencing data.
- Alternative splicing identification: This artifact uses exon-exon junctions predicted from paired-end RNA sequencing data as an input.

From a feature perspective, a significant pruning was performed. From all the features listed in Table 11 and the three types of analyses considered, gene expression analysis occupied the entire research and many features were not present in the end deliverable. Indeed, it was observed that the gene expression analysis step had a “jungle” of pre-processing methods and that the data sets were not appropriate for *differential gene expression analysis as bias correction requires replicates for each sample*. “Outlier mining” was more indicated and relevant for the current investigation. For instance, the count data sets, even normalized, were not suited for statistical comparison between genes due to their lack of technical or biological replicates. The only comparisons that could be obtained were purely exploratory and required additional investigation of the characteristics of the samples (like stage or control/tumor sample) to be informative.

Additionally, our research is focused on *reproducibility* and the concepts or practices falling under *reproducible systems* were also surprisingly diverse and sometimes richer than expected. All in all, a balance between the usefulness of the product artifact for biological research and the described in the literature to make computational research reproducible. It lead to a smaller set of features but they were anchored in relevant.

5.3 Problem understanding

The data sets used for gene expression analysis are products of Next-generation sequencing (NGS) protocols and instruments. This is important to note as the gene expression analysis files used here are not generated by micro-arrays for reasons that are out of the scope of this thesis. Nevertheless, we will see in the “pre-processing” sections that the methods that can be applied (as R packages for instance) are not all specifically designed for RNA-Seq data (NGS) or attempt to reproduce methods designed for micro-array analysis on RNA-Seq data.

Also, we are not dealing with reads, sequences or “low-level” data. Instead, the matrices we are manipulating are quite “high-level”, i.e. the product of a pipeline with multiple intervening algorithms. Here we explain how the data is generated, as it is part of the problem understanding. Only a succinct overview of how counts are generated is provided here. The pipeline is deliberately simplified as there

[15] Not every week in practice.

is no need of great details to understand what kind of biases are corrected or what the counts represent.

5.3.1 Pipeline BAM to counts

RNA from cells (tumor or healthy) is “converted” to DNA to be sequenced, in the lab. With RNA-Seq only a small part of the human genome is sequenced, the part being “expressed” at a given moment in time and of a specific group of cells (e.g. tumor cells). What we basically get is a blueprint of elements encoded in the genome (protein coding genes but also noncoding (RNA) genes) that are transcribed from DNA to RNA. All these transcribed elements in a cell are called the *transcriptome*. Hence, it can be said that the goal of RNA-Seq is to get the blueprint of the *transcriptome*.

The measure of counts started by one “conversion” already to prepare for sequencing, which is specific to RNA-Seq. Next, *transcriptome* assembly proceeds by rebuilding the blueprint from all these small sequenced fragments (called sequencing reads). But to achieve this, it will use a reference genome as a proxy for the transcriptome (Wang, Gerstein, & Snyder, 2009). With this alignment, we can deduct where a read fits on the genome (localization). Below an example is given with small transcripts fitting a protein coding gene. Figure 21 illustrates this. What we see is a gene represented with nucleotides (each letter) colored in black. Then, reads are represented by these small sequences on a white background. The match is represented in orange. Basically, this alignment information is stored in SAM files (or BAM their binary version) (Li et al., 2009). Next, *annotatio*n puts a label (like the gene id and gene name) on top of information contained in SAM/BAM files.

With the help of additional tools, a counting is performed per gene and sample. What is counted is the amount of reads, as seen in Figure 21 that overlap a gene. Here, we obtain a total of 8 reads and one question that could be asked is if for one or more samples there are counts that are much higher than others.

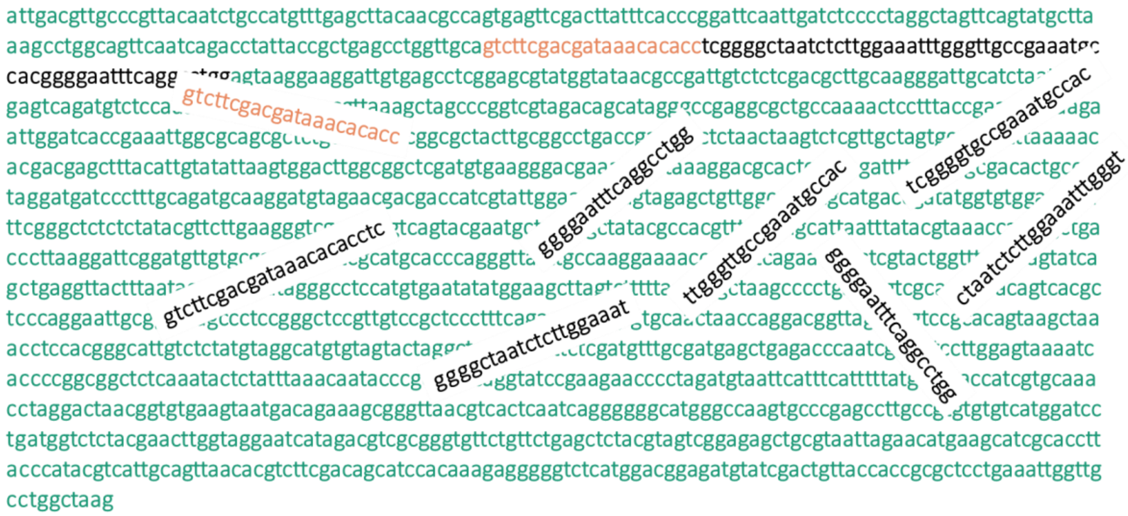


Figure 21 where reads are aligned to a reference genome

As another example, we could think about this gene sequence in black as being a goal, like the one found on a football field. Using this SAM/BAM file and a second file describing features, additional tools[16] will count the number of reads that corresponds to a gene. Just like someone would count the

[16] Again, plenty of them with their specificities… How do you count overlapping reads for instance?

number of balls left in the goal when the person is evacuated on a stretcher after a terrible scene of torture inflicted by this customized version of penalties.

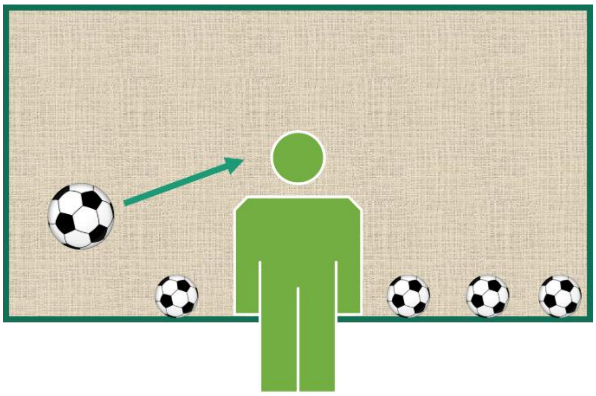


Figure 22 we shoot balls at a man until he cannot handle it anymore

Figure 23 shows a situation where there are more balls in one goal than the other one. If we still state that this goal is a gene, and that we want to know if the second one is more expressed, strictly considering this number might yield the conclusion that it is indeed the case. The second one looks more expressed… But maybe the gene is just longer than the first one. It may capture more reads which is not an indicator of any level of expression. That's a bias, as if we would need to know that there were two goal keepers in the second goal. This obviously necessitates more balls to get them on their respective stretcher[17] but is not an indication of their lower or higher strength to resist targeted penalties.

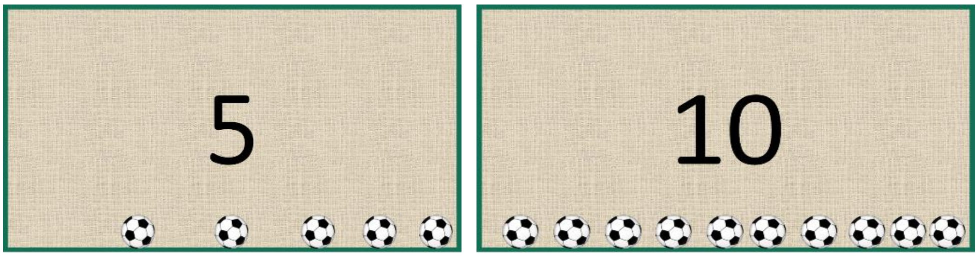


Figure 23 are these two situations really different?

Needless to highlight that there are many aligners, their choice addresses also what type of analysis is done with NGS. For RNA-Seq, except their velocity, aligners are handy if they handle alternative splices for instance, like STAR (Dobin et al., 2013). Alternative splicing is best visualized with colored balls from our goal. Figure 24 shows what is happening. Here between the two different splices, in the case of B we see an element that is missing (the orange ball). This modifies the sequence but it still originates from the same region of DNA and has therefore to be aligned appropriately. Some aligners manage to make that biological phenomenon visible with RNA-Seq data.

[17] If the basic dude resistance is set to a minimum threshold of more than two balls, in the face, per individual, let's be precise here!

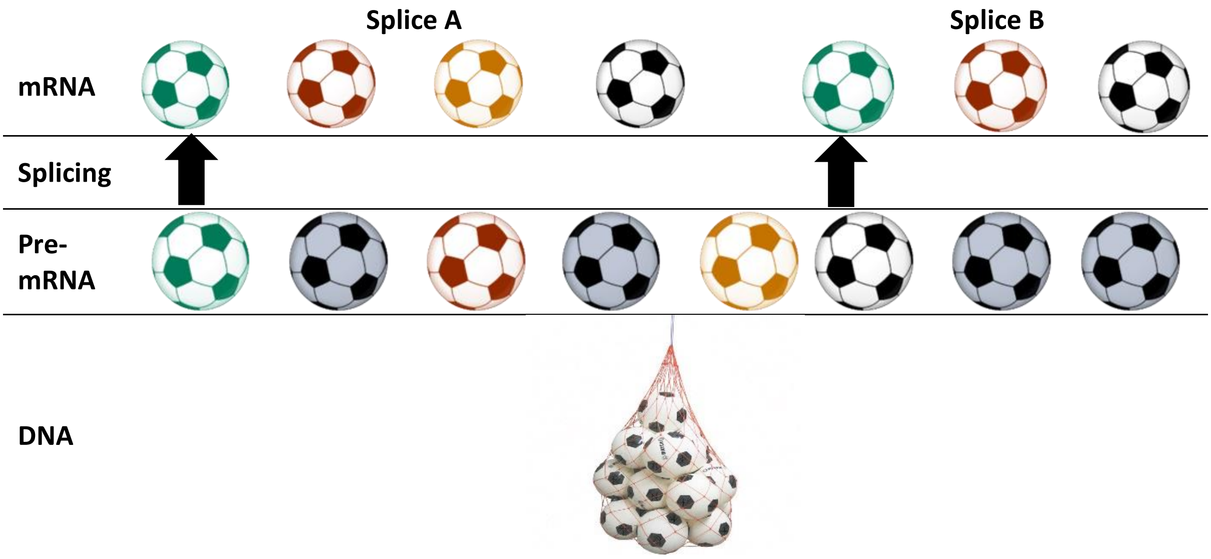


Figure 24 Illustration of alternative splicing

This presentation of tools intervening in a pipelines and the choice left to researchers to customize and adapts pipelines illustrates that what looks like an easy and standard process to obtain our counts is a perception far from reality. At the end of the day, we end-up with counts that need some normalization, i.e. find the “true” (or an approximation of the) biological variation cleaned from all types of technical artifacts, noise and make them comparable.

5.4 Pre-processing

We have seen that normalization was required to at least correct for variable sequencing depths (SD) across samples, to enable comparison between them. But why is a straightforward division of counts by the sum of all counts (total counts – TC/sample) not sufficient in case of *differential gene expression* analysis (DE)? What happens to the counts when we apply an R method from a package like edgeR or DESeq for another purpose than gene DE, like in our case, an outlier detection algorithm? What is meant by DE is a type of computational experiment searching to identify genes that have lower or higher expression between one or more treatments across the samples. For outlier detection, we assume that all samples are under identical conditions (all controls, all tumors, all treated or non-treated etc.). A pure simplification for convenience which is only tolerated because of the exploratory nature of our analyses without an assessment of whether features are *significantly* differentially expressed between two (or more) conditions. In the following paragraphs normalization methods for DE are described, that were initially applied on colorectal cancer RNA sequencing data.

5.4.1 How do results of a DE look like?

To make the contrast between DE and outlier analysis, an extract of DE analysis results is given in Table 12. What has been done is that one a data set of GEO (GSE33113) a TOP 250 genes analysis was executed with GEO’s geo2R interactive tool.

ID	ADJ.P.VAL	P.VALUE	T	B	LOGFC	GENE.SYMBOL
227140_AT	3.48E-19	1.03E-23	13.3645905	42.850153	5.8588524	INHBA
212942_S_AT	8.36E-16	4.95E-20	11.6033447	34.760715	5.0372944	CEMIP
225520_AT	4.03E-15	3.58E-19	11.1998529	32.864683	2.5062482	MTHFD1L
211253_X_AT	6.90E-13	8.17E-17	-10.1021361	27.651753	-4.5462945	PYY
213407_AT	1.04E-12	1.66E-16	-9.9592192	26.969056	-1.9957906	PHLPP2

Table 12 Example of results of DE

The R package running behind the scenes is Limma and the tool even offers a feature to get the R code interpreted on the fly. The difference with the outlier analysis lays in (1) the fact that we are comparing tumor samples versus normal samples in the data set and (2) that a DE test provides a measure of interestingness with a p-value. The null hypothesis states that a gene is not differentially expressed. Counts in two or more groups of samples are not different after correction of biases. Basically DE will evaluate differences in two groups (more complex designs are also possible with GLM features) whereas our outlier analysis just yields deviating data points per gene without any grouping per treatment, type of tumor etc…

5.4.2 An entry point: classification of "normalization methods" and biases

RNA-Seq gene expression analysis is also more precisely referred to as quantification of relative transcript abundance, as indicated by Pachter (2012).He categorized quantification models according to their genericity, complexity or for specific models how sequencing techniques (single-end, paired-end) are accounted for in a given model. Other criteria are sequence biases, i.e. bias due to chosen priming and fragmentation strategies in library preparation protocols. Sequence biases are mentioned by Hansen, Brenner, & Dudoit (2010) in the case of random hexamer priming. For instance, Random Hexamer (laboratory level) is a technique that might be applied in the amplification phase (creating a large numbers of copies of the same DNA, starting from a very low amount of DNA) at the very beginning of a sequencing step on the lab side that might introduce biases. What must be retained here is that there is, of course, a strong dependency between the lab and the files processed further in the bioinformatics pipeline. The biases that are corrected have their origin in the current knowledge in Biology, Sequencing technologies and statistics (a combination, not purely statistical procedures).

Additionally, events appearing to happen at random during the preparation steps might actually be biased, i.e. under certain conditions, like with neighboring sequences of fragments. Some of them may have an increased likelihood to be sequenced which impacts the expression counts and challenges the assumption of randomness (A. Roberts, Trapnell, Donaghey, Rinn, & Pachter, 2011). The same authors list three types of bias correction induced by "library preparation" and sequencing technologies. Their main hypothesis states that there is a general decrease of the differences between estimates after correction of the following biases and sub sequentially a decrease of genes "flagged" as differentially expressed when corrected for:

- **Technical replicates**: are "the sequencing of two different libraries that have been prepared using the same protocol from a single sample" (A. Roberts et al., 2011, p. 7) . Reduction of different expression estimates from two distinct libraries.
- **Library preparation methods**: Biases specific to preparation protocols (lab part).
- **Sequencing platforms**: e.g. Illumina or Solid. Biases specific to sequencing platforms.

As can be seen, what appears to be a "jungle" of methods available is actually due to the RNA-Seq technique itself. Third generation technologies (TGS) attempt to address biases caused by ambiguously mapping reads by sequencing greater read lengths (long reads) which may cover an entire RNA molecule and by sequencing native (non-amplified or unprocessed) RNA molecules (Dobin et al., 2013; Pachter, 2012; R. J. Roberts, Carneiro, & Schatz, 2013). This illustrates of course the awareness of these issues, and also the "depreciation" of current models on data generated with newer sequencing technologies.

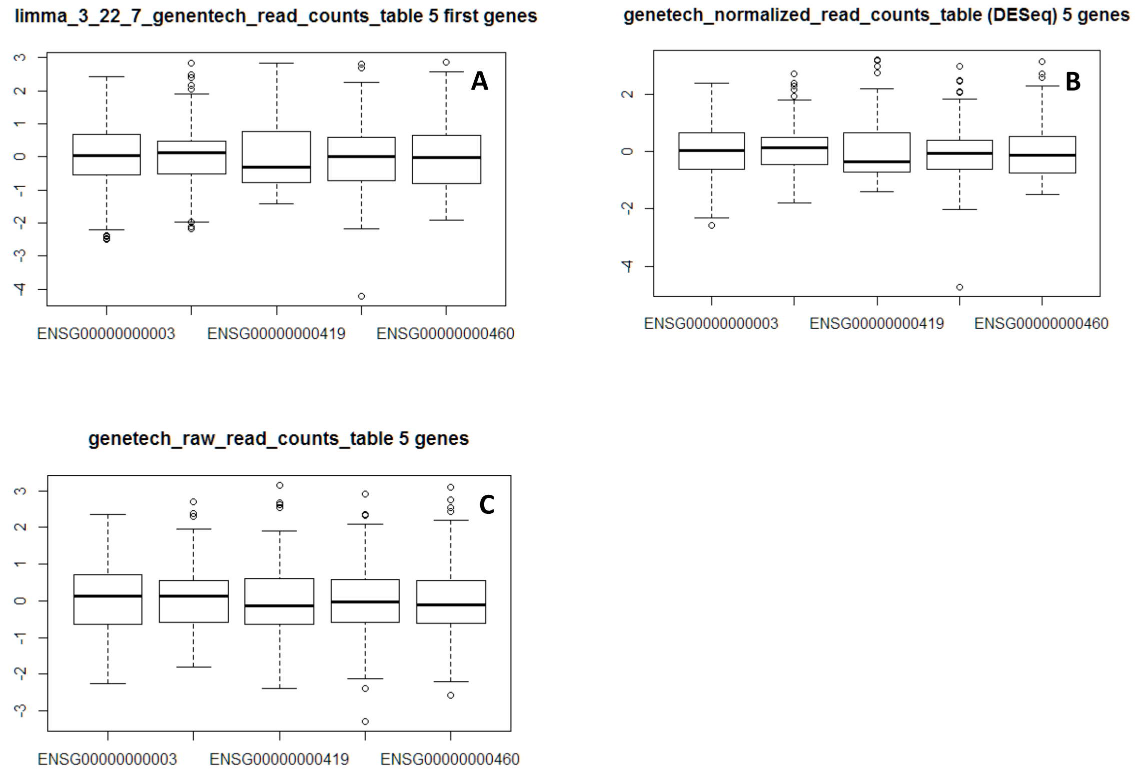


Figure 25 Boxplots of A) Normalized with Limma B) DESeq and C) Raw counts

A quick look at the normalized and standardized values on Genentech data set in Figure 25 which shows standardized log transformed normalized (A and B) and raw (C) data. Log transformation on raw and DESeq data computed a *pseudo count* = (count + 0.5). We see that outliers are behaving quite differently between each case.

5.4.3 Standardization

Before the clustering is executed, the normalized data may be scaled (z-scores). This might be a burden for the interpretation of the data. For instance a standardized range is not as informative as the normalized range which gives an idea of the real difference between two data points in units mastered by the end-user. It is also to note that in our case, standardizing the data had no impact on the output and or the distance value. Nonetheless, data standardization (and the fact that one has to apply it or not) is a crucial aspect related to the statistical methods that are used and the kind of output that is yielded to the end-users. Therefore, standardization is part of the “assumptions” and research context that would have to be shared. For instance, a recent study concluded that standardization does have a beneficial effect on the RNA-Seq data analysis when used with regression techniques (Zwiener, Frisch, & Binder, 2014).

5.4.4 Missing data

The current data sets (UMCU and Genentech) used have no missing values. Nevertheless, the state of future RNA-Seq technologies (which may be able to separate between the fact that no reads were mapped versus something went wrong during the sequencing step and attribute a value like N.A for instance) is assumed to be not known. Hence, a statement was added to remove the samples that contain missing values with the method *dropna*[18] of a pandas data frame. An equivalent approach can be applied in R directly with packages like edgeR by setting a parameter “na.rm” to TRUE in *calcNormFactors* function*,* for instance.

Missing data pre-processing is done before normalization, clustering and also plotting. In the last case, not controlling for missing data might create problems with the plotting libraries. The same holds for data with insufficient variation and some chart builders (like the *heatmap builder*) which will not render the expected visualization in that case.

Once more, missing data and to a certain extent standardization are important notions that should be described in the research context of a research object as they are part of the problems of the data analysis process. Even if this research mainly used “ideal” data sets with no missing values and required no standardization per se, this is not a commonly encountered situation in the KDD process applied in other contexts and data.

5.5 Data mining and Evaluation

5.5.1 Mining highly expressed samples per gene

This short chapter illustrates the outlier detection “algorithm” that was implemented to reproduce the following pattern:

- Low counts for most of the genes
- High counts for a number of samples between 1 and x (where x is an arbitrary number of samples, typically small or a percentage of the total number of samples present in a file). This is summarized by the following simple condition: $0 < \sum o_g <= x$ with o being a data point in the minority class of gene g.

Technically, we are not finding outliers in the sense of a data point which strongly deviate from the others or from any assumed probabilistic distribution to which the data would belong to. Still, when gathering requirements the denomination went from highly expressed genes to outliers. For this thesis, we indicate this by the more adapted notion of mining highly expressed samples for a given gene even if the screenshots of the tool present “outlier mining”. “Outlier mining” was kept to maintain communication between the “IT side” and the biological side understandable. The notion of highly expressed samples do not contradict the terms “ outlier mining” as one sample present in the data point is at least what we can call an outlier. To achieve this, a clustering-based algorithm was implemented, primarily for files normalized with discrete counts (i.e DESeq and edgeR) that are not log transformed.

The implemented algorithm is based on K-Means which is also adequate to detect outliers (Hodge & Austin, 2004). The only things that were added are a distance measure between two data points in each of the two (k=2) clusters detected by K-Means. Together with the range and support (column samples) it provides a quick overview of the situation without plots. A scatter plot might confirm to the user that the pattern is the one he is looking for.

5.5.2 A TOP-100 analysis

Here, we take a look at the two tables (based on a sample data set). The only difference is the normalization method applied (both with default parameters and no replicates). These methods are DESeq (or RLE) and edgeR (or TMM). We can assume that there will not be a perfect match between the two lists, but to what extent? We note also that no biological relevant information is considered

[18] http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.dropna.html

for this “evaluation”, it is therefore not concluded which list is better than the other but what the overlap is between these two lists.

To begin with, the lists were sorted by distance. Genes from DESeq match the edgeR top 100 list in 76% of the cases (3/4). The other genes appear later in the DESeq list (they have a lower distance). Only one gene (ENSG00000271369 - RP11-350D17.3) could not be matched at all (not present in DESeq list) but appears as a top entry in edgeR (and which explains why the correlation is done on n=99). Besides, 96% of the genes (n=99) have a similar support (number of samples detected as outliers). This will be described later and illustrated in Figure 31. The calculations are made on rounded distance values (4 numbers after decimal point).

The two following tables show the differences between DESeq and edgeR normalized data sets and the impact on the top list of outliers (based on distance between two data points assigned to two different clusters and which is expressed as a proportion of the range of the data). What can be seen is that in general, DESeq has a bigger influence on the range. The estimated counts (normalized) are shrunken. It is explained by the fact that DESeq uses the median for its estimation of size factors. edgeR has its own TMM (trimmed Mean of M-Values) method (the method applied by DESeq has the name RLE – Relative Log expression in edgeR). But again, other biases (as previously described) may or may not be taken into account and influence the normalized count versus the raw count to be higher or lower. Here we observe the tendency of edgeR to amplify the range, SF stands for Scaling Factor, even when RLE is used. DESeq appears to be inverted and reduce the range (except when edgeR reduces the range, then DESeq actually increased it (SF 5). The data set we refer to is umc_raw_counts.

Call edgeR:

```
calcNormFactors(object, method=c("TMM","RLE","upperquartile","none"),[…])
```

Figure 26 scaling factors with edgeR

Call DESeq:

```
estimateSizeFactors( object, locfunc=median, ... )
```

Figure 27 scaling factors with DESeq

Method	SF 1	SF 2	SF 3	SF 4	SF 5
edgeRR TMM	1.1242796	1.1124591	1.0642887	1.1064203	0.9300531
edgeR RLE	1.0616907	1.0802048	1.0377462	1.0940322	0.9443383
DESeq	0.9248139	0.6246225	0.7243243	0.6230260	1.0988168

Figure 28 Comparison of scaling factors edgeR (with RLE and TMM) and DESeq

We see that on the same data set (UMC RAW COUNTS) the first 5 size factors estimated by edgeR (both with TMM and RLE) are impacting the counts in another direction than what DESeq does.

ID	NAME	TYPE	SAMPLES	DISTANCE	RANGE
ENSG00000254656	RTL1	protein_coding	1	0.9970703125	1024.0
ENSG00000104827	CGB	protein_coding	1	0.996815286624	628.0
ENSG00000068985	PAGE1	protein_coding	1	0.99501867995	803.0
ENSG00000189052	CGB5	protein_coding	1	0.99427480916	3144.0
ENSG00000160181	TFF2	protein_coding	1	0.992922120642	132243.0
ENSG00000262117	BCAR4	lincRNA	1	0.991769547325	243.0
ENSG00000188984	AADACL3	protein_coding	1	0.990566037736	106.0
ENSG00000125255	SLC10A2	protein_coding	4	0.990285367335	6588.0

ENSG00000241168	RP11-10O22.1	lincRNA	1	0.988372093023	86.0
ENSG00000128564	VGF	protein_coding	1	0.987481945113	31155.0

Figure 29 edgeR (by distance) (applied to edgeR_3_8_6_genentech_read_counts_table)

The case that is illustrated here, is that a different normalization method yield a slightly different list of outliers. As we have seen, normalization methods are abundant, while their choice for exploratory data analysis is less strict than for a statistical test. One can imagine that someone having as information that the normalization method is RLE but uses it with edgeR, the list will, again, be slightly different (just by looking at the scaling factors). So, methods are linked with their original instantiation, or implementation to be replicable which confirms that only extracting methods from packages would be as limited as providing a package without the goals of the researcher.

ID	NAME	TYPE	SAMPLES	DISTANCE	RANGE
ENSG00000104827	CGB	protein_coding	1	0.998281786942	582.0
ENSG00000254656	RTL1	protein_coding	1	0.996466431095	849.0
ENSG00000189052	CGB5	protein_coding	1	0.995580110497	2715.0
ENSG00000188984	AADACL3	protein_coding	1	0.990825688073	109.0
ENSG00000068985	PAGE1	protein_coding	1	0.990243902439	410.0
ENSG00000118271	TTR	protein_coding	2	0.989599014643	7307.0
ENSG00000160181	TFF2	protein_coding	1	0.988602050437	83787.0
ENSG00000230198	RPL37P4	pseudogene	1	0.9875	80.0
ENSG00000128564	VGF	protein_coding	1	0.987474354821	27783.0
ENSG00000262117	BCAR4	lincRNA	1	0.986607142857	224.0

Figure 30 DESeq (by distance) applied to DESeq_1_18_0_genentech_read_counts_table

Figure 31 shows that there is some consistency between the two lists but we must be attentive to the units on the axes. DeSEQ has a larger variation (from 0.65 to 0.99 included) and edgeR starts at 0.94. This just shows that high distances in edgeR may be found later in the DeSEQ list when sorting the list on distance only.

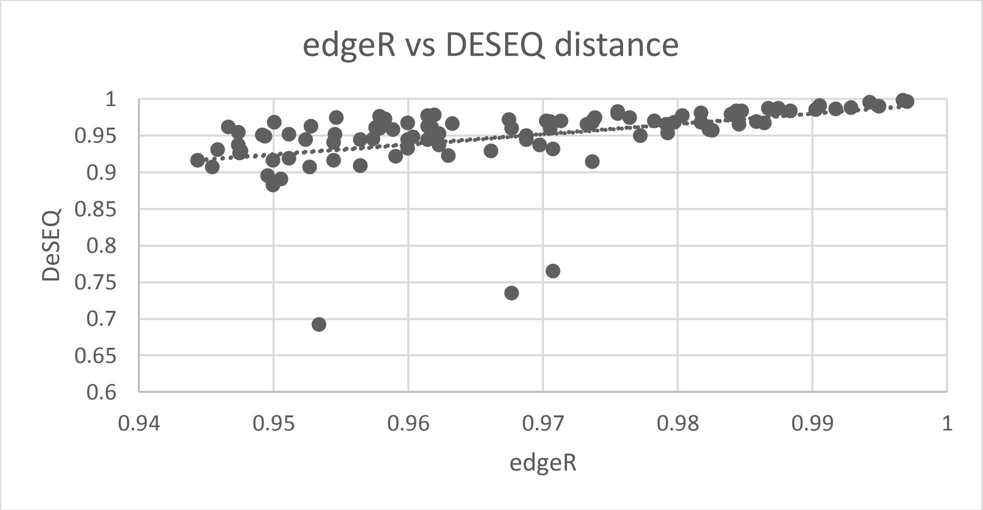


Figure 31 Top-100 genes. DESeq – edgeR comparison

The correlation between the edgeR data and the DESeq data is moderate ($\rho = .43,\ R^2 = .185, p < 0.01$) with n=99 but performs slightly better if we remove the three outliers (< 0.8 on the Y axis). Without these outliers, we obtain a stronger correlation ($\rho = 0.71, R^2 = .5, p < 0.01$) with n=96.

Interestingly, because the two normalization methods affect the range of the data differently the KMEANS clustering might yield a different "support". Moreover, the data sets remain quite different because of these "extra" normalization rules, like information shared between genes and calculation of average, mean or median, as can be seen from different scaling factors that are yielded although the chosen method (RLE) is the same. It also shows that in both cases, the distance is identical and therefore a "robust" measure (see Appendix A). The three outliers shown in Figure 31 are actually genes that have their samples clustered differently which affects the distance measured between the max data point in the minority class and the max data point in the majority class. In general, the fact that for our own data , edgeR is more sensitive (i.e. based on mean statistic) to outliers does not contradict the description of this model given by Anders et al. (2013).

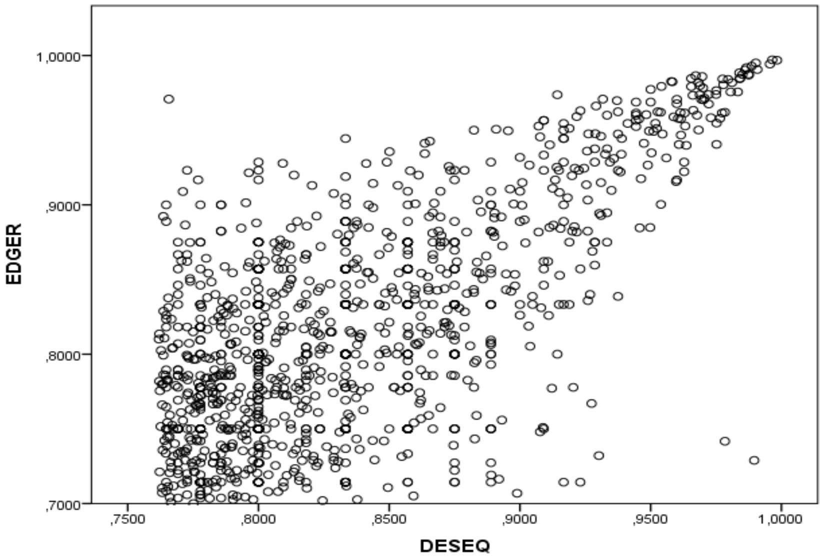


Figure 32 A comparison between edgeR and DESeq counts

It is to note that the number of non-matched genes present in an extended (n=2269) DESeq list but not in edgeR is quite important (42% - 956 out of 2269). This is partially explained by the fact that the edgeR normalized outlier list generator had a cutoff at 0.7 and DESeq at 0.6, so more edgeR genes were filtered out because of a too low distance and that's just an arbitrary decision. The second part of the explanation lays in alternative clustering (data points that are far apart in edgeR may become closer in DESeq and cluster differently) which lower the distance and pushes the gene lower in the top list.

To sum up, we might conclude that an acceptable overlap exist for distances greater than .95 but that the "confidence" in the distance drops quickly. Again, distance is not absolute and should be used together with the range of the data. The combined attributes offer a quick highlight on the interestingness of the outliers. As such, the algorithm in place is not really useful if it is considered as a strong outlier flagging algorithm but it becomes worthy if seen as a "high relative expression for some samples" pattern finder. This forces the end-user to be able to filter the list and reorder it on attributes that are not only the distance.

The two other attributes are support (i.e. number of samples in the minority class) and range (difference between max and min data point in the data set for one feature). Figure 33 illustrates how these properties look like on a Cartesian coordinate system. The advantage is that we let KMeans do its work while giving indications to the end-users about the coordinates of the outlier versus the rest of the data. As has been told previously, the filtering is not strict to not reject genes that might have a potential utility to appear in the outlier list.

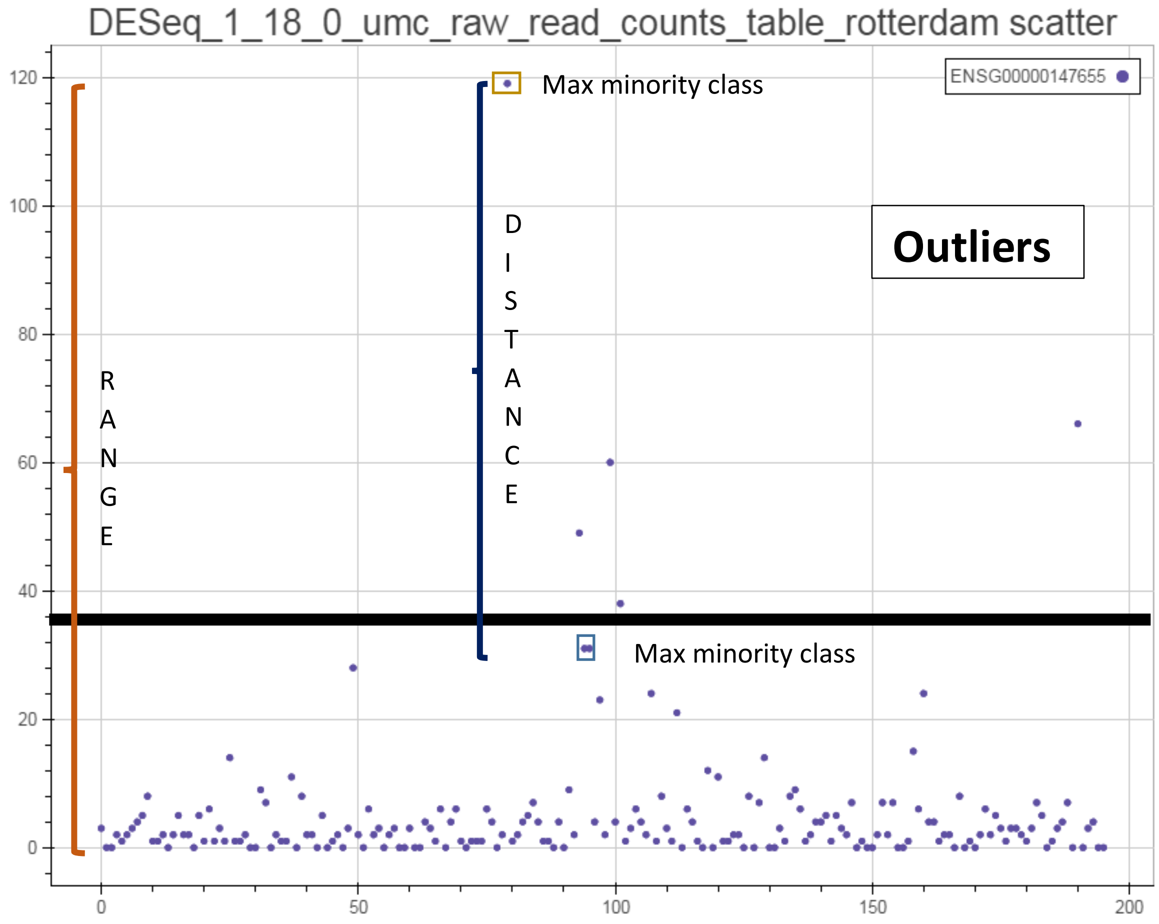


Figure 33 Export of the web-version of the scatterplot showing the results of RSPO2

5.6 Results Exploitation

The system built manages calls to python or R code transparently for the end-user. Bindings between R and Python are done to illustrate how one could potentially use the original implementation of a code chunk from another environment. As the results of our survey confirmed, there is a wide variety of programming languages or platforms used in bioinformatics. At the same time, RR aims at guarantying that the original code is executed (as they may be different implementation of a method, see DESeq and edgeR RLE). Hence, the end-user is manipulating different environments from its dashboard.

Results are transferred to the website to grids and plots (on demand). Grids offer filtering and ordering features. They also enable the user to select a sub list of features and plot them on scatter plots or heat maps. The main charts available are scatter plot, boxplot and heatmap. The scatter plot offers the option to dynamically show a sample label when approaching the mouse from a data point.

The first design cycle separated visualization from the rest of the KDD process. After focus group sessions a “second” design cycle reformatted everything to a real single page where visualization is activated on demand.

Chapter 6 The prototype: Implementation of Web Resources

Now, we depict the architectural and technical elements of the prototype that was implemented. More insight is given on how the concepts of transformation, resources and representations proper to the web architecture look like in a python web application. Also, some current frameworks (like AngularJS) in JavaScript for building modern web apps are presented. They particularly suited for a dynamic generation of content. The code of the prototype is hosted on Github under MIT license and is available there: https://github.com/armell/RNASEqTool.

6.1 Functional software architecture

In this section we present the main aspects of the functional architecture of the RNASeqTool. Minor adaptations were implemented after feedback was collected by our focus groups and our survey as a start of a second design cycle. Nevertheless, these adaptations do not fundamentally modify the description of the architecture provided below. These features provided a good start for discussing important or trivial elements of interactive data analysis for biologists.

6.1.1 Map of the website

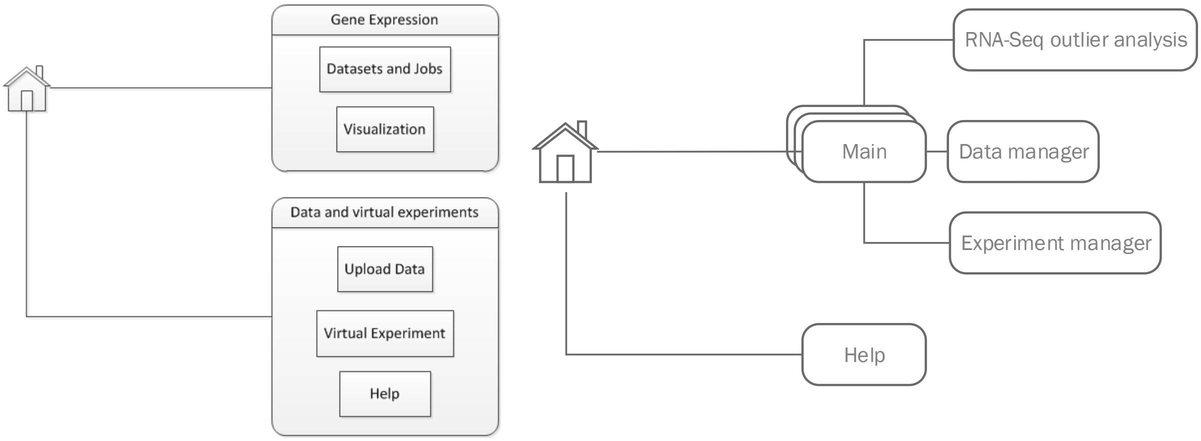


Figure 34 Structure of the evaluated user-interface (left) and updated version (right)

Figure 34 depicts the old (tested version) organization of the web application and how the pages were organized differently after feedback sessions with focus groups. Figure 35 is a use case of the different types of interactions a user could perform on the evaluated version. As can be seen, Data set selection and normalization are done inside a virtual experiment that has to be loaded and will render the appropriate methods. Based on a selected data set, a job can be configured and then executed to get a list of outliers. Retrieve genes displays a list of genes and metadata when the ID used in the data set enables it.

The major switch consists in a grouping of data set and visualization in one single page called RNA-Seq outlier as respondents preferred to have visualization integrated with data analytics without having to go to visualization in the menu. Additionally “upload data” and “virtual experiment” where renamed to “data manager” and “experiment manager” to provide more consistency and also indicating the real capabilities of the function. For instance, data manager groups uploads features and data set search and download.

6.1.2 Use case

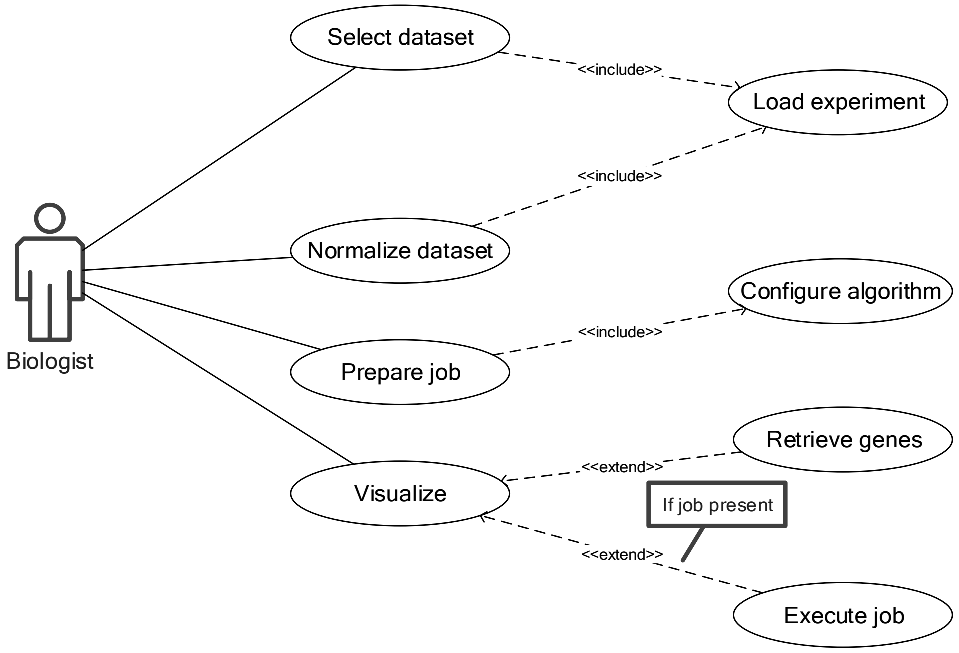


Figure 35 Use case diagram for a biologist (evaluated version)

The workflows remained similar across versions, before and after evaluations. Table 13 shows a mapping between the HCI-KDD steps elaborated earlier and the use cases in Figure 35. The scenario is the following: Data sets a grouped in experiments where a goal is defined. An example of a goal is “outlier analysis”. Then, a list of existing data sets appears with information about the last method applied in the pipeline. Here we can imagine a package or tool used to count the reads as explained earlier. If a “raw” data set is selected, it can be normalized or transformed via the four packages that we have encountered in Chapter 4.

HCI-KDD STEP	USE CASE
INTEGRATE	Load experiment and select data set
PRE-PROCESS	Normalize data set
MINE	Prepare job
VISUALIZE	Interaction (HCI)

Table 13 Mapping use cases to HCI-KDD steps

On normalized data sets, a biologist has the choice to apply a mining algorithm. A demo algorithm was implemented as outlier mining for count data (briefly explained in Chapter 5). By entering two parameters: minimum distance and maximum samples in outlier class a user *create a job*. Execute or run was avoided on purpose to demonstrate how a connection to a cluster could be explicitly presented to an end-user. For an end-user, the execution environment is loaded transparently. Here, mining was in python and normalization in R as showcase for multiple environment and reusability of packages.

Once a job is prepared it is marked as scheduled. The user had to select “visualization” from the menu, which strongly maps to HCI as a separate step. From there, jobs and existing data sets available in the currently loaded experiment are runnable. Selected an existing job allows to relaunch an analysis with identical parameters.

6.2 Technical Architecture

Because of the necessity to be able to ask different representations of different resources in a pure “compendium” style way of design an architecture, the web architecture (as a REST interface) was a first-class choice to implement our tool. Furthermore, uncountable projects of data interoperability and exchange, even in the fields of *omics make intensive use of REST interfaces. All the components added to our technical architecture where tested during the *ex-ante* evaluation, which is in our case simply what every well-intentioned developer does with new libraries, i.e. trying. The only exception is the name *ex-ante*, which makes it sound viciously serious and scientific.

Development has been done on a Linux virtual private server hosted by OVH. Operating system is Linux Ubuntu 14.10 (kernel 2.6.32). Python 2.7.8 and R 3.1.1 for data processing and analytics. Flask 0.10.1 for the REST API and AngularJS 1.3.15 as JavaScript framework (single page app). MySQL 5.5.41 is the relational database (with methods and experiments).These components enable data management with tools similar to what is used by bioinformaticians at the biomedical genetics department to facilitate the deployment on their servers. The main development environment was IntelliJ 14.1 Premium under academic license (free for non-commercial purposes).

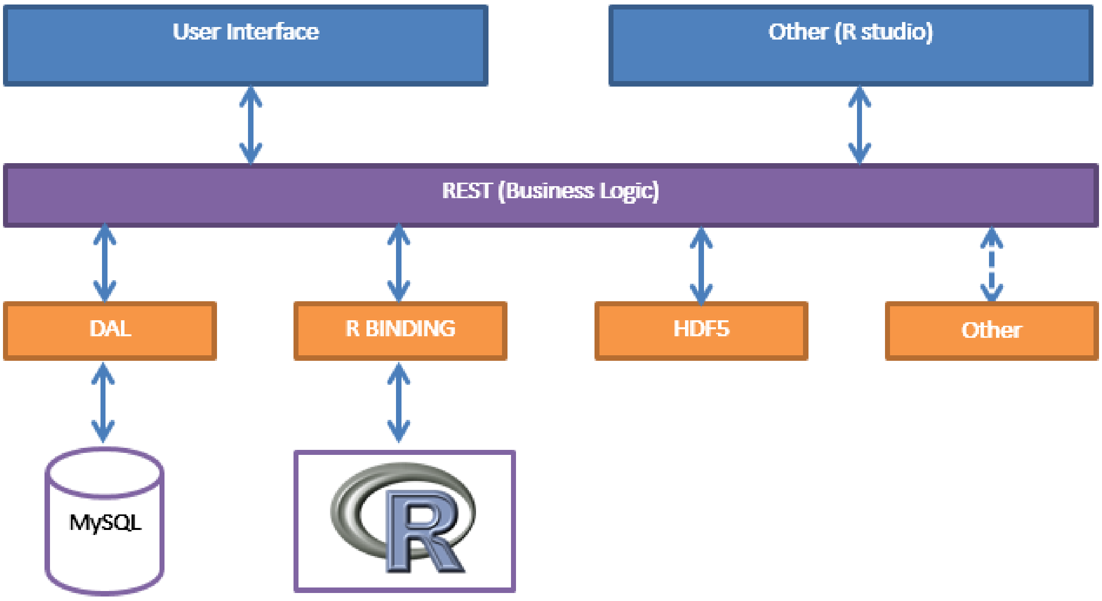


Figure 36 High level architecture of the application

6.2.1 DAL

Queries to MySQL were executed by an object-relational mapping library (Peewee) in python. Instead of writing SQL queries and instantiating classes manually, this is what the ORM does. It also manages relational constraints (foreign keys, many2many relationships…). Entities where queried through functions written in python which are part of the data access layer (DAL). Figure 37 shows a class that is mirroring one of the database tables in python.

```
class Packages(BaseModel):
    added_on = DateField()
    description = TextField()
    language = CharField()
    public_identifier = CharField()
    reference = CharField()
    url_source = CharField(null=True)
    version = CharField()
    name = CharField()

    class Meta:
        db_table = 'packages'
```

Figure 37 An entity class with the peewee ORM library

One advantage is to have directly python objects to work with and that an ORM helps to be relatively database independent (working with providers that are exchangeable). But as this is not a crucial choice as we do not discuss optimizations or how to query a relational database in python, the description is ended here.

6.2.2 R Binding

R binding is described in Chapter 4. In our technical architecture R binding works as a *data source* but is actually a bit more than that as R does some data analytics on data frames. Because we still have to deal with reproducible research and that package and version are a must-have, we can collect this information by introspection in python.

```
def DESeq_gene_expression_normalization(df_data):
    rpy2.robjects.pandas2ri.activate()
    df_data = df_data.dropna()

    r_data_set = robjects.conversion.py2ri(df_data)
    base = importr("base")
    DESeq = importr("DESeq")
    bio_generics = importr("BiocGenerics")
    rdiv = robjects.r.get('/')

    conds = base.factor(base.c(base.colnames(r_data_set)))

    cds = DESeq.newCountData set(base.round(r_data_set), conds)
    res_est = bio_generics.estimateSizeFactors(cds)

    normalized = base.t(rdiv(base.t(bio_generics.counts(res_est)),
bio_generics.sizeFactors(res_est)))
    rpy2.robjects.pandas2ri.deactivate()
    res = Result()
    res.frame = pd.DataFrame(numpy.round(numpy.matrix(normalized)),
index=normalized.rownames, columns=normalized.colnames)
    res.package = "DESeq"
    res.version = DESeq.__version__

    return res
```

Figure 38 How to call R code and being in a python environment at the same time

The solution shown in Figure 38 is shows how we interact with R via python. R objects are imported in python and it can go as far as the division ("/") symbol that must be imported from R as the standard "/" from python will not be interpreted correctly. Indeed, the python division does not know what to do between two R data frames. So the "/" from R is converted into an rdiv function in python… Not ideal but it's the true R code from DESeq which will be called inside an R session.

The last lines retrieve the version of the package used, R kindly sends this information back to python and it makes it as easy as asking the version of a "standard" python module.

6.2.3 HDF5

We are essentially working with pandas 'data frames in python. To accelerate and compress storage and retrieval of these data sets (quite frequent to retrieve genes or make plots) a secondary storage is used [19]. Pandas' optimized data frames combined with this storage and it increases the responsiveness of the app as there is no need to read a CSV file from a hard drive anymore but its binary representation in an HDF file.

In the data set table of our database, a mapping between public_identifier of the data set and its internal_identifier in HDF is present. Data sets in the HDF file are identified by a globally unique identifier (GUID) which is different from the public identifier for reasons that might be close to simply security and maintainability. The original data storage might be updated without touching the public_identifier.

6.2.4 REST

To build the architecture following the REST principles (Fielding & Taylor, 2000), FLASK and an additional package flask-Restful were adopted. An explanation of what REST and RESTFUL design constraints are can be found in numerous books like Pautasso (2014). In short, REST defines a uniform interface and identified elements called *resources*. The uniform interface is ensured by the primitive HTTP verbs (GET, POST, PUT, DELETE…) that are mapped to actions, respectively retrieve, create, update and destroy a resource.

Additionally, the payload (or *representation*) might follow structured formats like the collection+JSON format[20]. The partial implementation of this format served as a mockup for research objects as it implements the REST interface as an aggregation of resources and a single resource (for individual manipulation) as illustrated below. This is fine as we simulate what the behavior of an RO-based end-user interface might be and already check how they should be designed to provide sufficient information to users.

```
api.add_resource(expression_resources.Experiments,
"/api/expression/experiments")
api.add_resource(expression_resources.Experiment,
"/api/expression/experiment/<string:experimentId>")
```

Figure 39 adding resources to the api with Flask

In Figure 39, a resource class is linked to an URL. Here it means that to get a list of experiments the URL will be http://domain.tld/**api/expression/experiments**. An example of a class answering to a request made on the URL above is given in Figure 40.

```
class Data sets(restful.Resource):
    def get(self):
        data sets = dr.get_all_data sets()
        transfer_data sets = []
        for d in data sets:
            transfer_data sets.append(eto.Data setView(d).to_json())

        return cr.JsonResource(transfer_data sets)
    def post(self):
        [...]
```

Figure 40 representation of a (collection of) resource(s) with a python class

Last, any resource can potentially send one or more representation (if supported). For instance, summarized data as JSON or the original CSV content with a 'text/csv' mime-type. This is shown in Figure 41 with a configuration of different outputs using *Flask* and *Flask-Restful*.

[19] http://www.hdfgroup.org/

[20] http://amundsen.com/media-types/collection/format/

```
@api.representation('application/json')
def output_json(data, code, headers=None):
    resp = make_response(data.to_json(), code)
    resp.headers.extend(headers.items().append({"Location":
request.base_url}) or {"Location": request.base_url})

    return resp

@api.representation('text/csv')
def output_csv(data, code, headers=None):
    strbuffer = StringIO()
    data.to_csv(strbuffer, index=False)
    resp = make_response(strbuffer.getvalue(), code)
    resp.headers.extend(headers.items().append({"Location":
request.base_url}) or {"Location": request.base_url})

    return resp
```

Figure 41 configuring multiple representations of a resource with Flask

We have seen of these abstract concepts of representation and resources are implemented in our tool. This implementation makes it possible to add an extra representation, for instance, which would be the research objects representations. For now, we work with JSON without semantically enriching the content.

6.2.5 User interface

The first choice that was tested is the Ember.js framework. It was found out that too many conventions (a position defended by the designers of ember) were more of an issue to develop our user interface. An example is the strict JSON format that was expected by the data layer of ember. Hence, the angular.js framework was selected and offered all the help needed to build our single page app.

From angular, at least two important features were needed. Asynchronous calls to our REST API and data binding. Asynchronous calls enable a better responsiveness of the app which may change of state while a task is ongoing (like outlier retrieval). So, the user is not blocked on a task that may be really data-intensive. Outlier mining lasts a few minutes but the analytics performed have nothing in common with bigger or higher dimensional data sets that may be processed in other steps of RNA-Seq or other fields. A good example is High-Throughput screening (HTS) data analysis with heavy computational tasks and gigabytes of high-dimensional data (Omta, Egan, Spruit, & Brinkkemper, 2012).

Asynchronous tasks offer, at least a sensation, of a better responsiveness as the app is not blocked on a single task but the results might still take some time to appear. A small adaptation was implemented under the feature “outlier streaming” which exploits HTTP streaming capabilities. Again, this does not make it faster but gives the user immediate indication that the task is ongoing or that preliminary results can be checked as soon as an algorithm identified an interesting outlier.

The second aspect is data binding. With angular.js, data is stored in a variable called *$scope* and all the interactions with the user interface and the model ($scope) are processed by angular. For instance, clicking on a data set in the virtual experiment section will automatically update a variable “selected data set”, totally locally without any call to a remote server or extreme JavaScript plumbing. It is also convenient to build templates in an HTML document, like a list of methods. *Directives* (special angular tags, starting with ng-*) read the content of a JavaScript collection (like an array) and inject appropriate code in the DOM.

```
<md-list ng-repeat="mm in infoMethodsMine">
    <md-list-item>
        <div class="md-avatar" flex="10">
            <md-button class="md-fab md-mini md-accent" aria-label="FAB"
                       ng-click="selectMiningMethod(mm)">
                <span class="glyphicon glyphicon-eye-open" aria-
hidden="true"></span>
            </md-button>
        </div>
        <div class="md-list-item-text" flex="90">
            <table layout-padding>
                <tr>
                    <th>Package</th>
                    <td>{{mm.data.name}}</td>
                </tr>
                <tr>
                    <th>Version</th>
                    <td>{{mm.data.version}}</td>
                </tr>
                <tr>
                    <th>Language</th>
                    <td>{{mm.data.language}}</td>
                </tr>
                […]
        </div>
    </md-list-item>
</md-list>
```

Figure 42 HTML code with angular directives

The *md-** tags are from *angular-material*. This extra module provides basic building blocks for an application that follows Google's material design guidelines. The sequence diagram below shows which technical elements are involved in a transaction from experiment selection, gene filtering and html plotting Figure 43.

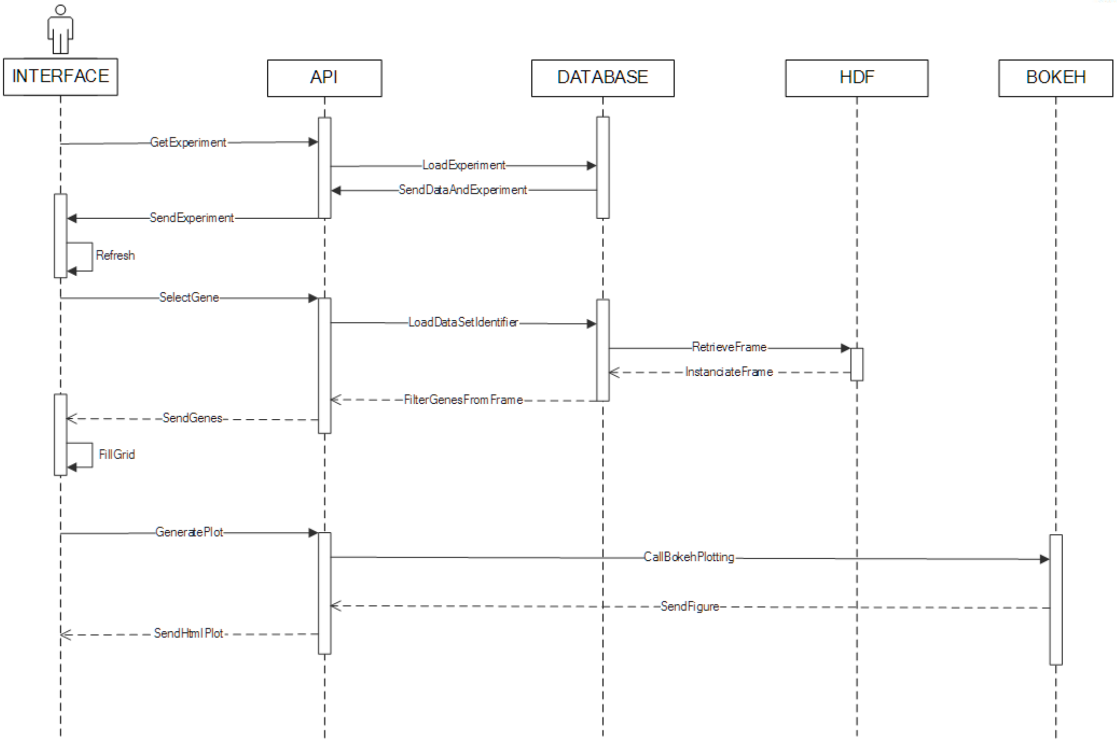


Figure 43 Sequence diagram of html plotting

6.2.6 And how it looks for a bioinformatician

A bioinformatician might not want to access the data sets created by biologists via the GUI interface. Therefore, a direct to the API simplifies data retrieval directly in R studio for instance.

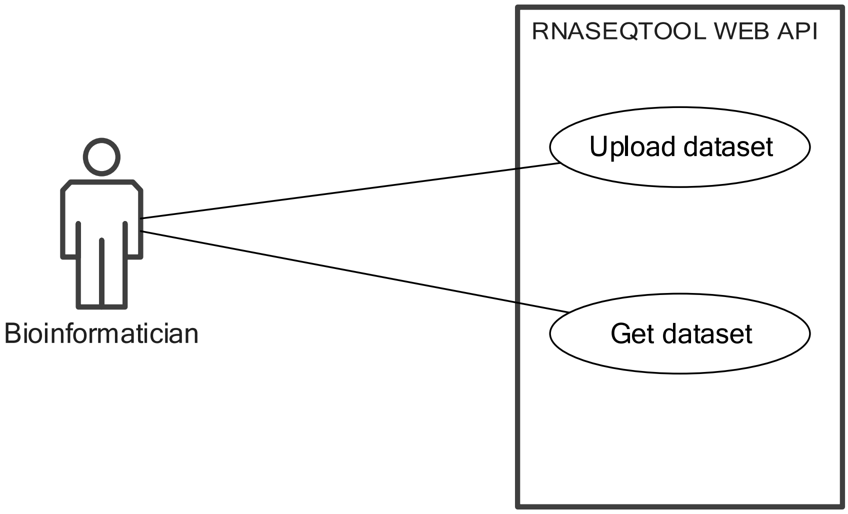
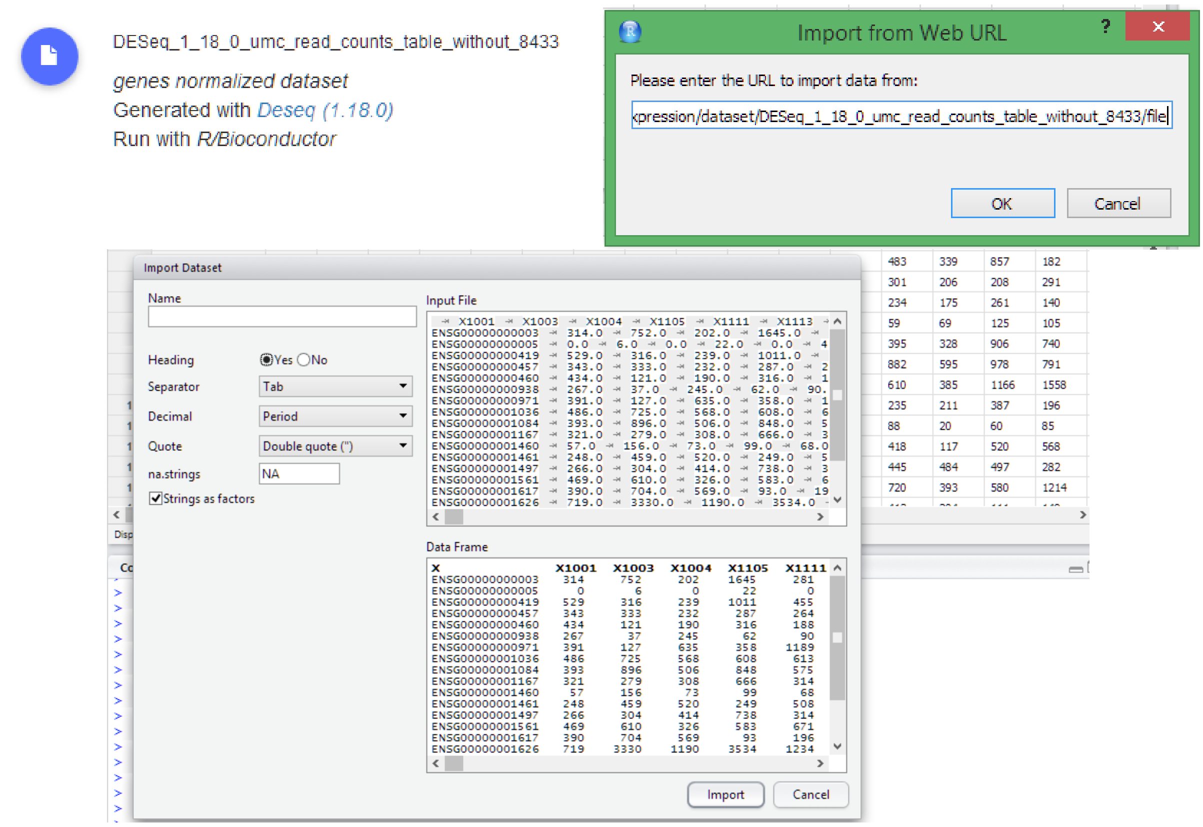


Figure 44 Bioinformatician example use case

Figure 44 shows a use case of a bioinformatician opting for a Web Api, where data upload and retrieval are made possible. Knowing the public identifier of a data set normalized by a biologist, it's easy for him to import it directly in R (if that's his preferred language). Figure 45 depicts the situation where a normalized DESeq data set is selected and entered in the "Import from Web URL" option in RStudio. Then, the data appears in TSV format with samples and genes. It also ensures that it's the correct data set on which alternative data analytics will be made on.



DESeq_1_18_0_umc_read_counts_table_without_8433

genes normalized dataset

Generated with *Deseq (1.18.0)*

Run with *R/Bioconductor*

Please enter the URL to import data from:

xpression/dataset/DESeq_1_18_0_umc_read_counts_table_without_8433/file

Name:

Heading: ● Yes ○ No

Separator: Tab

Decimal: Period

Quote: Double quote (")

na.strings: NA

☑ Strings as factors

Input File

```
 ⇥ X1001  ⇥ X1003  ⇥ X1004  ⇥ X1105  ⇥ X1111  ⇥ X1113  ⇥
ENSG00000000003 ⇥ 314.0 ⇥ 752.0 ⇥ 202.0 ⇥ 1645.0 ⇥
ENSG00000000005 ⇥ 0.0 ⇥ 6.0 ⇥ 0.0 ⇥ 22.0 ⇥ 0.0 ⇥ 4
ENSG00000000419 ⇥ 529.0 ⇥ 316.0 ⇥ 239.0 ⇥ 1011.0 ⇥
ENSG00000000457 ⇥ 343.0 ⇥ 333.0 ⇥ 232.0 ⇥ 287.0 ⇥ 2
ENSG00000000460 ⇥ 434.0 ⇥ 121.0 ⇥ 190.0 ⇥ 316.0 ⇥ 1
ENSG00000000938 ⇥ 267.0 ⇥ 37.0 ⇥ 245.0 ⇥ 62.0 ⇥ 90.
ENSG00000000971 ⇥ 391.0 ⇥ 127.0 ⇥ 635.0 ⇥ 358.0 ⇥ 1
ENSG00000001036 ⇥ 486.0 ⇥ 725.0 ⇥ 568.0 ⇥ 608.0 ⇥ 6
ENSG00000001084 ⇥ 393.0 ⇥ 896.0 ⇥ 506.0 ⇥ 848.0 ⇥ 5
ENSG00000001167 ⇥ 321.0 ⇥ 279.0 ⇥ 308.0 ⇥ 666.0 ⇥ 3
ENSG00000001460 ⇥ 57.0 ⇥ 156.0 ⇥ 73.0 ⇥ 99.0 ⇥ 68.0
ENSG00000001461 ⇥ 248.0 ⇥ 459.0 ⇥ 520.0 ⇥ 249.0 ⇥ 5
ENSG00000001497 ⇥ 266.0 ⇥ 304.0 ⇥ 414.0 ⇥ 738.0 ⇥ 3
ENSG00000001561 ⇥ 469.0 ⇥ 610.0 ⇥ 326.0 ⇥ 583.0 ⇥ 6
ENSG00000001617 ⇥ 390.0 ⇥ 704.0 ⇥ 569.0 ⇥ 93.0 ⇥ 19
ENSG00000001626 ⇥ 719.0 ⇥ 3330.0 ⇥ 1190.0 ⇥ 3534.0 -
```

Data Frame

```
X                 X1001   X1003   X1004   X1105   X1111
ENSG00000000003     314     752     202    1645     281
ENSG00000000005       0       6       0      22       0
ENSG00000000419     529     316     239    1011     455
ENSG00000000457     343     333     232     287     264
ENSG00000000460     434     121     190     316     188
ENSG00000000938     267      37     245      62      90
ENSG00000000971     391     127     635     358    1189
ENSG00000001036     486     725     568     608     613
ENSG00000001084     393     896     506     848     575
ENSG00000001167     321     279     308     666     314
ENSG00000001460      57     156      73      99      68
ENSG00000001461     248     459     520     249     508
ENSG00000001497     266     304     414     738     314
ENSG00000001561     469     610     326     583     671
ENSG00000001617     390     704     569      93     196
ENSG00000001626     719    3330    1190    3534    1234
```

483	339	857	182
301	206	208	291
234	175	261	140
59	69	125	105
395	328	906	740
882	595	978	791
610	385	1166	1558
235	211	387	196
88	20	60	85
418	117	520	568
445	484	497	282
720	393	580	1214

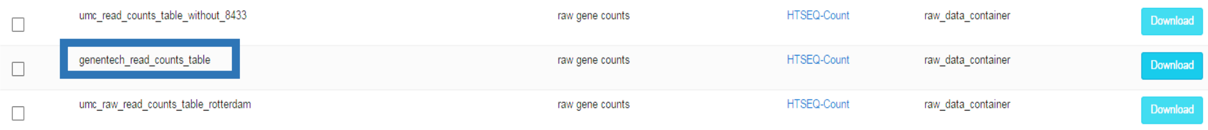
Figure 45 The data set has a public identifier and can be imported in RStudio easily

6.3 Public identifiers and database

6.3.1 Public identifiers for access: showcase

To answer the constraint of accessing components of an experiment, components had to be identified to be retrievable. What appears to be a simple task (identifying components) is actually one of the biggest challenges we noted during this work. Not only are data sets identified but also packages and methods.

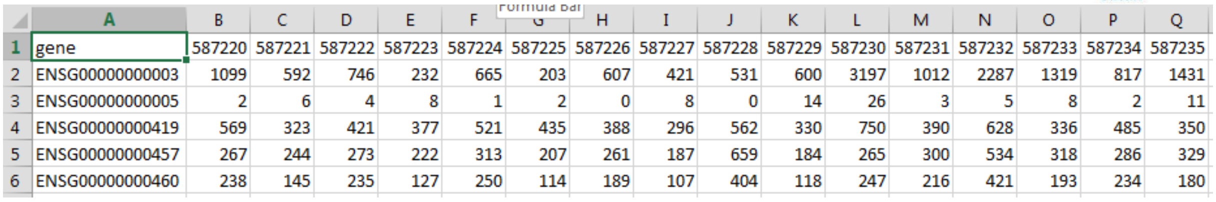
To discuss the role of identifiers for reproducibility omnipresent in our application, we show one case. First access to a data set via download in the browser.



□	umc_read_counts_table_without_8433	raw gene counts	HTSEQ-Count	raw_data_container	Download
□	genentech_read_counts_table	raw gene counts	HTSEQ-Count	raw_data_container	Download
□	umc_raw_read_counts_table_rotterdam	raw gene counts	HTSEQ-Count	raw_data_container	Download

Figure 46 Download from graphical user interface

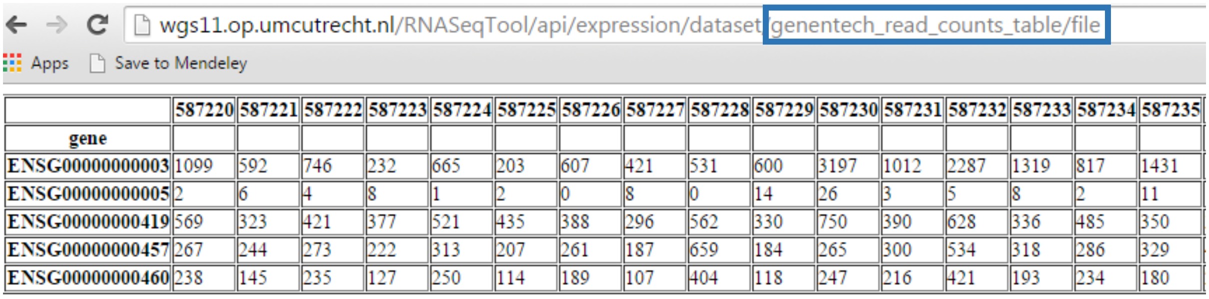
Figure 46 shows the list of data sets available in the application. A search function was added to filter the content of the grid in full-text. Queries such as "DESeq" yield all data sets normalized with this package. The element in a blue frame, "genentech_read_counts_table", is a raw data set identifier.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	gene	587220	587221	587222	587223	587224	587225	587226	587227	587228	587229	587230	587231	587232	587233	587234	587235
2	ENSG00000000003	1099	592	746	232	665	203	607	421	531	600	3197	1012	2287	1319	817	1431
3	ENSG00000000005	2	6	4	8	1	2	0	8	0	14	26	3	5	8	2	11
4	ENSG00000000419	569	323	421	377	521	435	388	296	562	330	750	390	628	336	485	350
5	ENSG00000000457	267	244	273	222	313	207	261	187	659	184	265	300	534	318	286	329
6	ENSG00000000460	238	145	235	127	250	114	189	107	404	118	247	216	421	193	234	180

Figure 47 TSV imported in excel version via download

Figure 47 is the excel file after importing the TSV file downloaded from the user interface. This does not require to know the public identifier in contrast to the second option where it becomes useful.

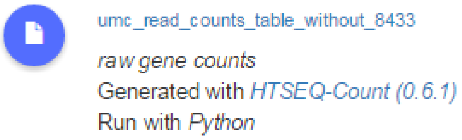


wgs11.op.umcutrecht.nl/RNASeqTool/api/expression/dataset/genentech_read_counts_table/file
Apps
Save to Mendeley

	587220	587221	587222	587223	587224	587225	587226	587227	587228	587229	587230	587231	587232	587233	587234	587235
gene																
ENSG00000000003	1099	592	746	232	665	203	607	421	531	600	3197	1012	2287	1319	817	1431
ENSG00000000005	2	6	4	8	1	2	0	8	0	14	26	3	5	8	2	11
ENSG00000000419	569	323	421	377	521	435	388	296	562	330	750	390	628	336	485	350
ENSG00000000457	267	244	273	222	313	207	261	187	659	184	265	300	534	318	286	329
ENSG00000000460	238	145	235	127	250	114	189	107	404	118	247	216	421	193	234	180

Figure 48 HTML version of data set directly from browser

Figure 48, the HTML representation can be used as a quick overview of the file from the web app. This overview is accessible for a user from the list of data sets in an experiment by clicking on the identifier. Figure 49 shows a public identifier of a data set and the information displayed to the user.



umc_read_counts_table_without_8433

raw gene counts

Generated with *HTSEQ-Count (0.6.1)*

Run with *Python*

Figure 49 the data set identifier is a link to an HTML view

To achieve this, we added some classes dedicated to transfer the right representation based on the accept header. Here we mainly show human representations but exactly the same principle would allow to send semantically enriched messages. Figure 50 presents python code which transforms a matrix into csv or html representations.

```
class DataFrameApiResource(ApiResource):

    def to_csv(self):
        str_buffer = StringIO.StringIO()
        self.content.to_csv(str_buffer, sep="\t")

        return Response(str_buffer.getvalue(), mimetype="text/csv",
headers={"content-disposition":"attachment; filename=\"" + self.name +
"\""})

    def to_html(self):
        print("html content")
        str_buffer = StringIO.StringIO()
        self.content.head().to_html(str_buffer)
        print("responding")
        return Response(str_buffer.getvalue())
```

Figure 50 Example of representation generation according to accept value

6.3.2 Looking at the tables

Figure 51 shows how tables are related to each other. The main tables are data sets (the count matrices) and experiments. Packages and methods were modeled as followed: one package may contain one or more methods. A package has a *running environment* and a *version*. Packages are seen as an *instantiation* of a method or set of methods. Implementation plays a role as we have seen with the edgeR/DESeq case where RLE (from DESeq) applied with edgeR will yield different results because of different modelization applied.

The other way is also a reality. Methods have many instances (implementations) that are using building blocks from other methods and models. This makes the implementation too simple but more maintainable. Here, there is just a link between methods and their implementation. This simple database design has already its drawbacks. Indeed, this would imply that information from *inside* the packages is extracted, similar to biomodels, i.e. model curation. This information could potentially not be limited to a scenario like selected edgeR for TMM but offer more hints about the assumptions. Still, the challenges to maintain a more complex data base design requiring subtle details about packages are great.

A ranking was added on methods to provide a notion of workflow but this does not appear in the tool. Ranking and information extraction would be part of *method recommendation*, a feature arising from our focus groups (see Chapter 7). Table 14 shows how the main tables are related. Here we simplified real case scenarios where many packages are applied on one data set. In this more complicated case, *custom-packages, a feature* examined by focus groups, would be extremely useful. Indeed, a single identifier of a custom-package may be applied instead of single identifier by package used.

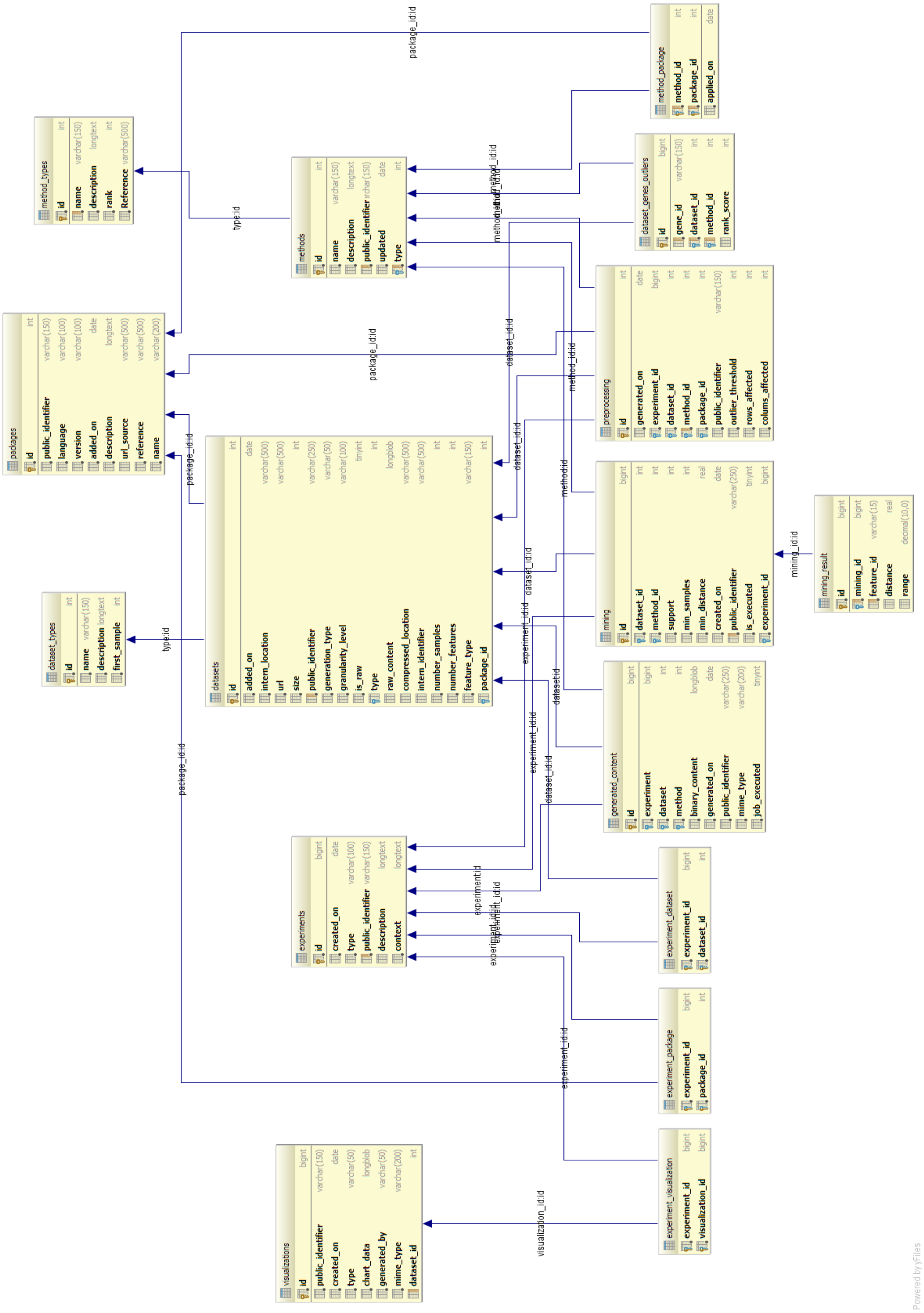


Figure 51 Database schema

TABLE NAME	DESCRIPTION	LINKS	TYPE
EXPERIMENTS	Make the link between the context of the experiment and data sets etc.	Data sets	Many-to-many
DATA SETS	Each time a data set is created it is linked to a specific package.	Packages	Many-to-one
METHODS	A method is like RLE or TMM a model based on assumptions that is applied on a data set.	Packages	Many-to-many
PACKAGES	A package may implement one or more methods	Methods	Many-to-many

Table 14 Main tables and description

6.3.3 Exchanged messages

As part of the URI of a resource, a “public identifier” is assigned to each data set, method, package and job. Public identifiers can be extended to other resources like charts following the same pattern.

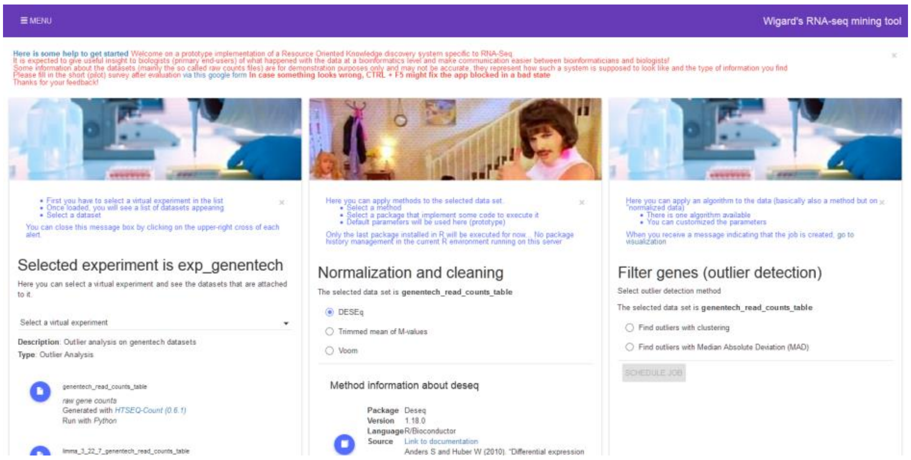
```
{
        "items": [{
                "href":
"http://example.com:8888/api/expression/experiment/exp_umc_outlier",
                "data": {
                        "public_identifier": "exp_umc_outlier",
                        "experiment_type": "Outlier detection in RNA-seq read counts
data",
                        "description": "Default experiment for UMC data",
                        "created_on": "2015-04-30"
                }
        },

        {
                "href":
"http://example.com:8888/api/expression/experiment/exp_tcga",
                "data": {
                        "public_identifier": "exp_tcga",
                        "experiment_type": "Outlier analysis",
                        "description": "Outlier analysis on TCGA data",
                        "created_on": "2015-05-01"
                }
        },
        {
                "href":
"http://example.com:8888/api/expression/experiment/exp_rotterdam_demo",
                "data": {
                        "public_identifier": "exp_rotterdam_demo",
                        "experiment_type": "Outlier analysis",
                        "description": "One
data set is loaded (originally umc data set).",
                        "created_on": "2015-05-26"
                }
        }],
        "collection": {
                "href": "http://example.com:8888/api/expression/experiments/"
        }
}
```

Figure 52 Example of json fragment for experiments

6.4 User web interface

6.4.1 Calling packages

As explained previously, for the end-user there is no R or python script to be aware of. Packages are added to the tool and appear in a list based on the selection of (1) main method in the first iteration (2) main goal in the second adaptation. Figure 53 shows that the methods are classified by model. Indeed, DESEQ or TMM are statistical models to process data. The choice of a package using this classification was not optimal and therefore updated to what can be seen in Figure 54.



Here is some help to get started Welcome on a prototype implementation of a Resource Oriented Knowledge discovery system specific to RNA-Seq
It is expected to give useful insight to biologists (primary end-users) of what happened with the data at a bioinformatics level and make communication easier between bioinformaticians and biologists!
Some information about the datasets (mainly the so called raw counts files) are for demonstration purposes only and may not be accurate, they represent how such a system is supposed to look like and the type of information you find
Please fill in the short (pilot) survey after evaluation via this google form **In case something looks wrong, CTRL + F5 might fix the app blocked in a bad state**
Thanks for your feedback!

- First you have to select a virtual experiment in the list
- Once loaded, you will see a list of datasets appearing
- Select a dataset

You can close this message box by clicking on the upper-right cross of each alert.

Selected experiment is exp_genentech

Here you can select a virtual experiment and see the datasets that are attached to it.

Select a virtual experiment ▾

Description: Outlier analysis on genentech datasets
Type: Outlier Analysis

genentech_read_counts_table
raw gene counts
Generated with *HTSEQ-Count (0.6.1)*
Run with *Python*

limma_3_22_7_genentech_read_counts_table

Here you can apply methods to the selected data set.
- Select a method
- Select a package that implement some code to execute it
- Default parameters will be used here (prototype)

Only the last package installed in R will be executed for now... No package history management in the current R environment running on this server

Normalization and cleaning

The selected data set is **genentech_read_counts_table**

● DESEq
○ Trimmed mean of M-values
○ Voom

Method information about deseq

Package Deseq
Version 1.18.0
Language R/Bioconductor
Source Link to documentation
Anders S and Huber W (2010). "Differential expression

Here you can apply an algorithm to the data (basically also a method but on "normalized data)
- There is one algorithm available
- You can customized the parameters

When you receive a message indicating that the job is created, go to visualization

Filter genes (outlier detection)

Select outlier detection method

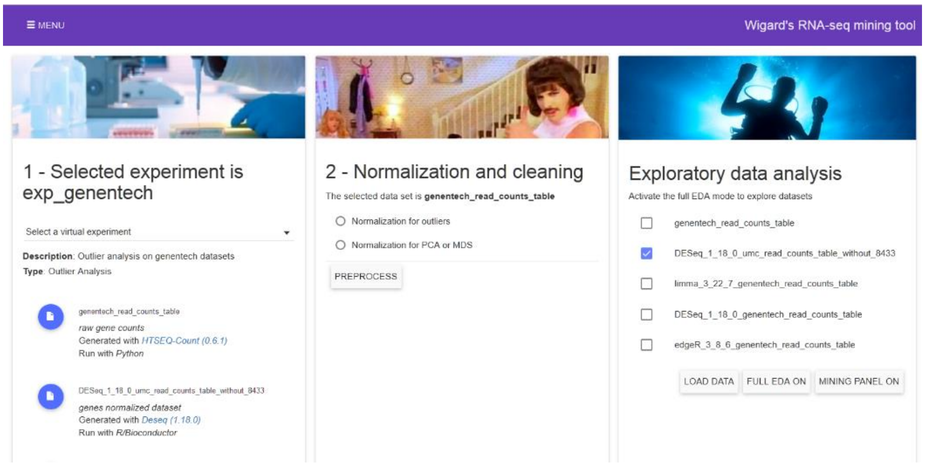
The selected data set is **genentech_read_counts_table**

○ Find outliers with clustering
○ Find outliers with Median Absolute Deviation (MAD)

Figure 53 First version with visualization and preparation (i.e. normalization) separated

6.4.2 Dynamic plots

Figure 54 is a screenshot of the second iteration of the website, after evaluation. Because of their priority, slightly better exploratory data analysis features (plotting and comparison) were added. The mining panel (filter genes in Figure 53) is now accessible by clicking on “Mining panel on”.



1 - Selected experiment is exp_genentech

Select a virtual experiment ▾

Description: Outlier analysis on genentech datasets
Type: Outlier Analysis

genentech_read_counts_table
raw gene counts
Generated with *HTSEQ-Count (0.6.1)*
Run with *Python*

DESeq_1_18_0_umc_read_counts_table_without_8433
genes normalized dataset
Generated with *Deseq (1.18.0)*
Run with *R/Bioconductor*

2 - Normalization and cleaning

The selected data set is **genentech_read_counts_table**

○ Normalization for outliers
○ Normalization for PCA or MDS

Exploratory data analysis

Activate the full EDA mode to explore datasets

☐ genentech_read_counts_table
☑ DESeq_1_18_0_umc_read_counts_table_without_8433
☐ limma_3_22_7_genentech_read_counts_table
☐ DESeq_1_18_0_genentech_read_counts_table
☐ edgeR_3_8_6_genentech_read_counts_table

Figure 54 Single page app last version

In Figure 55, two scatter plots showing the same data set but with two different normalization procedures. It simply renders what visual scenario analysis would look like. At the same time, we already see that data points may really differ in both cases. A label interactively appears on the right plot when the user approaches a data point with his/her mouse.

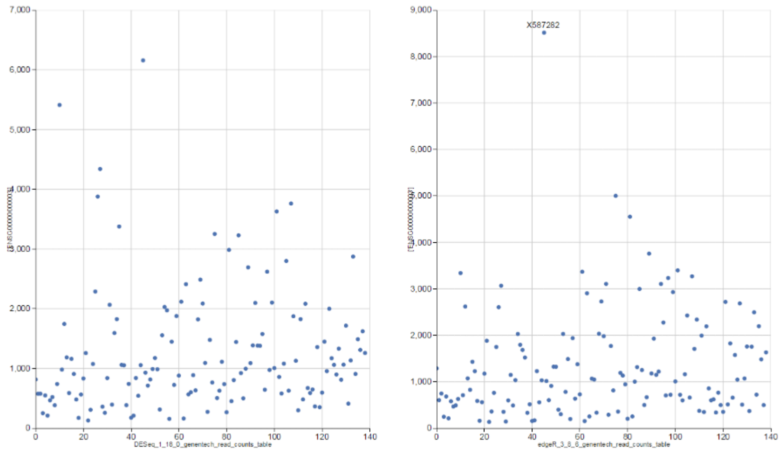
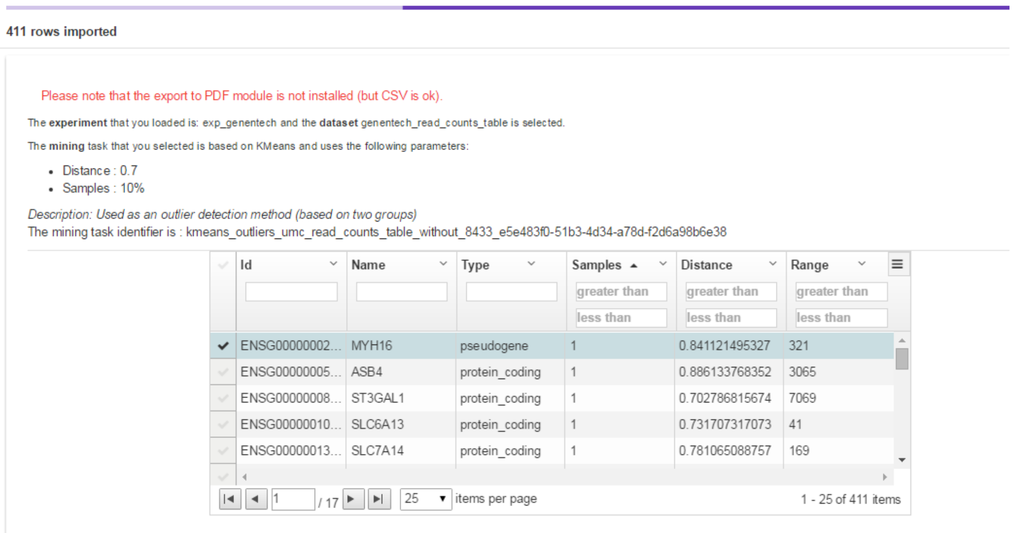


Figure 55 Dynamic plots with mpld3

As such, it provides a quick way to investigate the data of one gene on two different data sets and check if they have a similar number of outliers, for instance.

6.4.3 Filter genes

While outliers are popping out of the algorithm, the user can interact with a grid where genes appear by batch. Selecting one or more genes can be followed by visualizing a scatter plot or a heatmap. Figure 56 shows a previously created job that is reloaded an executed again with the same parameters (recorded in the database). An "export grid as CSV" option has been activated to continue the analysis with external tools.



411 rows imported

Please note that the export to PDF module is not installed (but CSV is ok).

The **experiment** that you loaded is: exp_genentech and the **dataset** genentech_read_counts_table is selected.

The **mining** task that you selected is based on KMeans and uses the following parameters:

- Distance : 0.7
- Samples : 10%

Description: Used as an outlier detection method (based on two groups)
The mining task identifier is : kmeans_outliers_umc_read_counts_table_without_8433_e5e483f0-51b3-4d34-a78d-f2d6a98b6e38

	Id	Name	Type	Samples ▲	Distance	Range
				greater than less than	greater than less than	greater than less than
✓	ENSG00000002...	MYH16	pseudogene	1	0.841121495327	321
	ENSG00000005...	ASB4	protein_coding	1	0.886133768352	3065
	ENSG00000008...	ST3GAL1	protein_coding	1	0.702786815674	7069
	ENSG00000010...	SLC6A13	protein_coding	1	0.731707317073	41
	ENSG00000013...	SLC7A14	protein_coding	1	0.781065088757	169

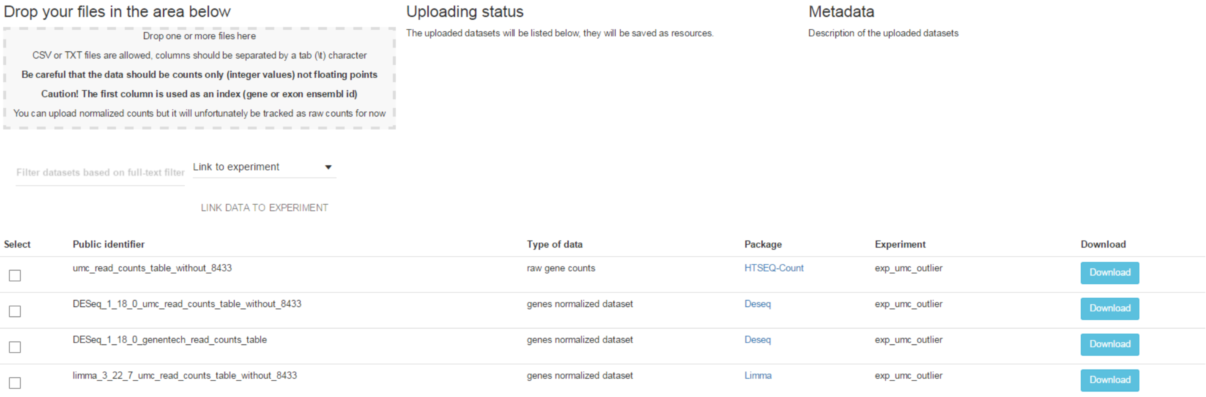
1 / 17 25 items per page 1 - 25 of 411 items

Figure 56 Filtering genes from a dynamic table and seeing previous parameters

All columns are sortable or searchable. This is to enable an easy classification of the output of the algorithm, also on two columns.

6.4.4 Data management

Last, a quick overview of the data management page where users can upload data sets by dragging and dropping gene count tables. When the data set is uploaded, meta-data can be entered (here limited to a public_identifier).



Drop your files in the area below

Drop one or more files here

CSV or TXT files are allowed, columns should be separated by a tab (\t) character

Be careful that the data should be counts only (integer values) not floating points

Caution! The first column is used as an index (gene or exon ensembl id)

You can upload normalized counts but it will unfortunately be tracked as raw counts for now

Uploading status

The uploaded datasets will be listed below, they will be saved as resources.

Metadata

Description of the uploaded datasets

Filter datasets based on full-text filter

Link to experiment ▾

LINK DATA TO EXPERIMENT

Select	Public identifier	Type of data	Package	Experiment	Download
□	umc_read_counts_table_without_8433	raw gene counts	HTSEQ-Count	exp_umc_outlier	Download
□	DESeq_1_18_0_umc_read_counts_table_without_8433	genes normalized dataset	Deseq	exp_umc_outlier	Download
□	DESeq_1_18_0_genentech_read_counts_table	genes normalized dataset	Deseq	exp_umc_outlier	Download
□	limma_3_22_7_umc_read_counts_table_without_8433	genes normalized dataset	Limma	exp_umc_outlier	Download

Figure 57 Data management for users

A table of existing data set gives information on the package used, the experiment it is linked to and an option to download. Above, a text field can be filled in to filter the table based on a full-text criteria. It is also a shortcut to get all the data sets normalized with “Limma” essentially. Next to the text field, a link to experiment option. This option makes possible the linking of experiments to data sets. So, this data set will be available in the list when the corresponding experiment is loaded.

Chapter 7 Evaluation

At the end of the first design cycle, the product artifact was evaluated. Experts from the UMC and Erasmus Medical Center (Rotterdam) discussed the design choices in focus groups. At the same time, a survey was sent. This survey covered aspects like the programming languages used, verification and understanding of previous work, use of reproducible research tools… A usability scale to get some insight about the HCI part of the process was also submitted. To answer this part, the tool was hosted on a server publicly accessible. Preloaded data sets were available and respondents could judge the interface. A supplementary open question asked for features that were really missing.

7.1 Qualitative: Focus Groups

In this section, we assess the suitability of the tool to tackle data analytics and reproducible research challenges. Having a working tool was precious for people attending these meetings as they could see (or for some manipulate) a real user-interface hiding quite complex architectural choices. But the advantages and limitations of the design choices and features added are has to be judged. For instance, what is its impact on the knowledge discovery process? This is the goal of confirmatory focus groups (Tremblay et al., 2010). Three sessions were organized. They gathered biologists and/or bioinformaticians to discuss design choices and their impact on the field. As introduction, participants were asked how this tool could be made more general in terms of methods available or data sets processed. All sessions were recorded after participants of focus group 1 and 2 gave their consent by signing a form with their names and roles in the organization.

The separation between “biologist” and “bioinformatician” is based on their primary objectives (more lab or data analytics) despite that this separation is not obvious and that some biologists are programming too. Simply put, this separation is used to clarify a situation and avoid the denomination “computational biologist”. This binary state makes it easier to discuss during a focus group but is not optimal and in accordance with a much more complex reality. Still, the discussions were distinctly oriented towards visualization in presence of biologists and concerns about accurate data manipulation with bioinformaticians

Although participants of both “categories” were invited for all the focus groups, the two last focus groups consist of bioinformaticians only. The perspective is therefore unilateral but nonetheless valuable to keep as feedback for further refinements of such a prototype and build a realistic design proposition.

After the pre-test, some modifications were added to the artifact. These modifications appeared as screenshots on slides and were a basis for elaborating if these new design choices improved the situation or not. So technically all the users accessed the same tool all the time to not modify the effect of the tool during the evaluation. Nevertheless, modifications are debated in the external and internal focus groups (like new classification of packages).

7.1.1 Pre-test

Organized on May 29 2015, number of participants 7.

IDENTIFIER	DOMAIN	ORGANIZATION
A1	Biology	UMC
A2	Biology	UMC
A3	Biology	UMC
A4	Bioinformatics	UMC
A5	Biology	UMC

A6	Biology	UU
A7	Biology	UMC

From the steps missing in the tool are quality checks. These quality checks are not feasible as count matrices files do not record these details that are captured much earlier in the pipeline. It means that we should have started with BAM files or raw sequencing reads. This is an indication toward a more complex data management part with more intervening methods for *biologists*.

For computational intensive procedures, the explicit presence of a “Job submission” system is acceptable. However, much more customization is needed for normalization or mining. Method comparison corresponds to a wish expressed by more than one participant. The only burden is that a large choice of methods implies a method recommendation system according to participants. Ideally automatically based on the input data (uploaded files). For these methods, a template could be prepared by a bioinformaticians (default values, best choice for one case…).

There is more need for visualization at every stage of the process (or certainly more EDA). Comparisons between methods is done by visualizing results with plots. The impact on a method could be judged based on a random sampling of few occurrences (samples or genes) in the data set. The biggest issue is how to have updated packages (i.e. methods) to analyze and compare data sets. For instance, a recently published paper suggest a new method to normalize a data set, the tool should be modular enough to just plug it in.

There should be a minimal integration about how the methods are implemented or suggested. So, not too much information on the screen as linked sources are convenient enough to navigate through. Generating a dynamic document with code and context is not seen as important, just nice to have.

7.1.2 Internal Focus Group

Organized on June 11 2015, number of participants 5.

IDENTIFIER	DOMAIN	ORGANIZATION
B1	Bioinformatics	UMC
B2	Bioinformatics	UMC
B3	Bioinformatics	UMC
B4	Bioinformatics	UMC
B5	Bioinformatics	UMC

The main problem with opening a data set is to know its story. What is the experiment and hypotheses? Also, the purpose of such a tool seems unclear. Is reproducibility about what someone did or to do something new? At first sight, it looks like a smart way of storing. Knowing what kind of packages were used and create scenarios. The tool should offer standardized options. Available pre-processing methods are useful to know the data and what the impacts are, interactively. Knowing how much information is lost depending on what kind of method was applied (e.g. after corrections) is important.

How to use these methods is an issue, e.g. what decisions should be taken. Also, merging data sets is suggested by a participant for meta-analysis of expression data. Starting with earlier data in the pipeline would also be handy. Another participant explains that it is actually someone who does not care about how the data was generated that should use this tool. In the pipeline, different actors are intervening. Someone looking at the BAM files is in a different phase than the person who analyzes the outputs. Also normalization is guided by the type of instrument used.

At least one of the participants did not use data tracking mechanisms but said that research done by just pushing on some buttons is not good. Prior knowledge is required (i.e. for the KDD process, ed.). There are different people intervening and it is a complex situation. There are the data out of the machines and the data to analyze. Here the users do not need to know the “dirty things” about the data but what is happening conceptually and the research questions. An analysis is not run without knowing what to do and what the parameters ‘values do.

Even if a bioinformatician prepare modules as a black box, how will it be interpreted? As much knowledge as a bioinformatician is needed, also knowledge about statistics. An improvement would be an addition of a visible workflow that might be edited on the fly, by replacing steps. In bioinformatics, scripts might be tailored to the data set at hand. An IPython document may give valuable indications and putting all the code open source is the way to go.

7.1.3 External Focus Group

Organized on June 18 2015, number of participants 3.

IDENTIFIER	DOMAIN	ORGANIZATION
C1	Bioinformatics	EMC
C2	Bioinformatics	EMC
C3	Bioinformatics	EMC

Participants see this tool as primary focused for biologists who want to do something with computers but who do not have sufficient expertise in programming. They also say that there is a need for that kind of tool. A general workflow is missing, i.e. Indications where to start and where to go. The second step (normalization) is not interesting for people just willing to watch results. The difference between DESeq packages is understood by bioinformaticians, biologists would not know which one to choose.

If a biologist want to normalize a data set, he should send it to a bioinformatician. On the user interface a default package should be shown and other packages could be listed if the user want to do so. But explanations from Bioconductor will not be read by biologists.

Reproducibility is a big problem because one does not know how these algorithms work and certainly when it is a data set coming from outside. Even more, sometimes a piece of software works stochastically. So, it cannot be 100% reproducible but good documentation (also about the hardware used) is a must.

Biologists don’t understand the packages and a bioinformatician just want to use the command line with parameters etc. Sometimes, the selection of packages is done because they are convenient and people are happy with them. Also, the “execute job” feature should be removed as it is again not suitable for biologists.

A method recommendation system should be based on knowledge of the experiment (context) and what someone wants to do with the data. For instance, fusion genes with outliers could be simply renamed fusion genes detection. Biologists do not bother if it’s normalization for outliers or PCA, it should be per goal rather than technique but still let the choice for some cases. Also, if there is contextual information and background about projects or public data then a search option is required.

Like "I want to search for breast cancer patients" in the data sets. The goal is to compare the data sets and see if there is a presence or absence of a gene for instance.

Then, a bioinformatician could check the suitability of the methods used because there are a lot of different solutions. It is not a standard investigation. The problem with the two step approach (normalization and mining, ed.) is that it represents two possibilities to make errors (false positives or so). Exploratory data analysis is more important for biologists as they can play with their data.

For the code, a link to GitHub would be sufficient. Such a website should be separated by type of experiment (with RNA-Seq or other, outlier mining etc.). Then, lots of visualization (not processing, ed.).

7.2 Quantitative: Survey

A survey was submitted by e-mail to bioinformaticians and biologists inside and outside the UMCU. This was done at the beginning of June, at the end of a first design cycle. First, general questions about Reproducible Research and knowledge discovery were asked, the survey ended with a few questions about the usability of the product artifact.

7.2.1 Respondents

The respondents were invited by email to answer a survey with an opportunity to consult the web application and give their opinion. Respondents were invited among the group of Medical Genetics or EMC (Rotterdam), by waves to avoid a crash of the web server, to answer a survey with a link to the tool. The response rate is 33% (11 out of 33 contacted people).

The total number of respondents is not high (n=11) but still offers a range of profiles that belong to this research domain and hence provides some form of representativeness despite its convenience sampling strategy. There was a total of 6 bioinformaticians and 5 biologists that answered the survey. Respondents have IT skills covering the whole spectrum of the self-reported IT level on a scale from 1 to 5 (see Figure 58). Biologists assessed their IT skills within a range from 1 to 3 whereas bioinformaticians expectedly reported higher IT skills within a range from 3 to 5. The level three was a prerequisite to use the command line and SSH which was the case for 2 out of 5 biologists.

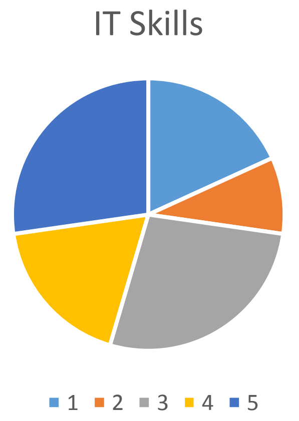


Figure 58 Level of reported IT skills

7.2.2 Analysis of Reproducible Research and Knowledge Discovery

Firstly, the responses of the survey suggest that there is a benefit to enable partial reproducibility for verification purposes. For two questions asking how frequently a full or partial replication would be needed from 1 (never) to 5 (always) the most occurring answer for full replication is low (Median=2, Mode=2) whereas it is higher for partial replication (Median=3, Mode=3). The results of the survey also indicate that respondents emphasize replication for verification (Mode=4) over understanding (Mode=1) of published results when it comes to partial replication. For understanding instead of verifying, partial replication has 2 respondents needing to perform it frequently. Full replication on the other hand has strongly negative answers (only values 1 – never – and 2).

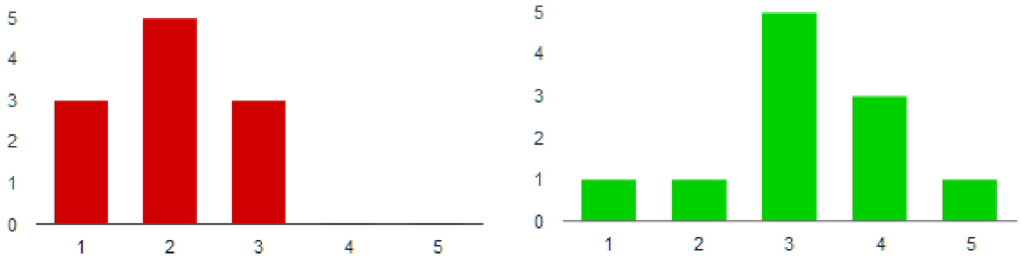


Figure 59 Results for full replication on the left on partial on the right

When it comes to considering a package as a black box biologists (median = 3, max=4) and bioinformaticians (median=3, max=3) indicate that, for two respondents at least, biologists might be more inclined to consider a package as a black box.

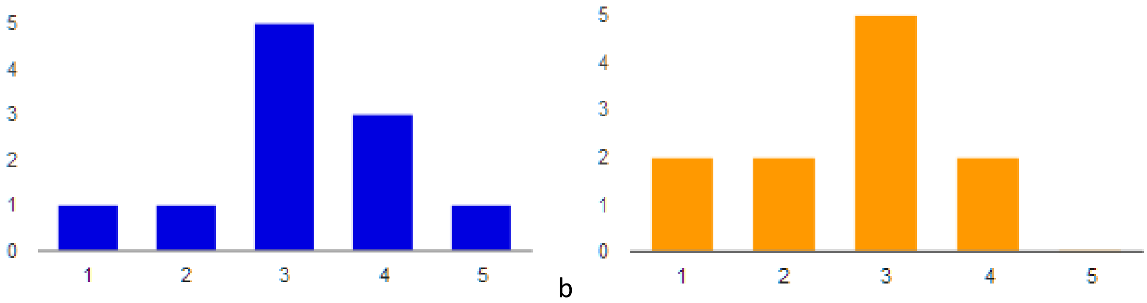


Figure 60 Hard to understand published data set (left) and own lab data set (right)

Figure 60 shows a subtle distinction about the fact that it is more frequently difficult to understand a published data set than from the lab. This is in accordance with the fact that the paper and supplementary material is rarely sufficient to have enough information about a study. An R package on the other hand is seen an appreciable benefit for biologists (Median=4, Mode=3) and bioinformaticians (Median=4, mode=4).

Finally, about practices in bioinformatics only one biologist reported not to use a programming language, at least for scripting. Most of them (60%) know python or R. All bioinformaticians reported to know at least R, then python and Perl for some (33%). Additional languages reported (other) are D, Stata and Visual Basic. Table 15 shows a summary of the results of languages selected in the choices offered to respondents.

LANGUAGE	#	PERCENTAGE
R	7	63.6%
PYTHON	6	54.5%
PERL	5	45.5%
JAVA/GROOVY	1	9.1%
C#	0	0%

BASH	5	45.5%
C	1	9.1%
C++	0	0%
NONE	1	9.1%
OTHER	3	27.3%

Table 15 Distribution of languages used

7.2.3 Usability – SUS Scale

The System Usability Scale (Brooke, 1996) was added to the questionnaire to get some insight from a pure end-user perspective on some design choices (induced by our architecture) which may need to be improved to make the tool convenient for daily usage or correspond better to the end-users' needs for interactive data exploration. This score helps us to answer our fourth research question as it gives some quantitative notion of the impact of architectural choices and end-users who do not focus on that kind of details. Enough evidence from the field was collected that such a system could be useful, but the other part, how usable it is at the moment also needed to be evaluated.

Based on a subsample of respondents (n=8) to the survey which had access to the website[21], a usability score was calculated. The average SUS score computed is 67 which puts the artifact close to the average (68) which corresponds to the lower extremity of a C grade (i.e. acceptable) according to the most common reference average (Bangor, Kortum, & Miller, 2009a). We also note that it was a task-free test, hence the scores are based on the real first impression. This score indicates that the product in its current state has a wide range of possible improvements. The focus groups offer more in depth knowledge about what seems right or wrong in the system at the knowledge-discovery side. Still, the SUS result was followed by one open question on what was really missing, as stated by respondents. Table 16 summaries these remarks. We note that not all respondents where familiar with the goals of the tool and its specificity which made it hard to review for some of them.

RESPONDENT	**SUMMARY**	**COMMENT**	**CORRESPONDING FOCUS GROUP OUTCOME**
A	More EDA integrated with the methods selection (not after in visualization)	"general plotting options during all stages of the data processing, including custom selection of subsets of the data"	More EDA
B	Combination of virtual experiment and separation between KDD steps to harsh	"I believe a more logic flow would be necessary. It was not intuitive to define the order to follow in selecting a data set, type of experiment, and go back to the menu to visualize... there was no immediate sense of order."	-
C	Too simple for bioinformaticians and too complex for biologists	"Leave the bioinformatical analysis at the bioinformatical desk. Biologists can't appreciate all the ins and outs of the steps you provide at the website. For example, do you	Method recommendation system Templates

[21] The usage of a WIFI connection inside the UMC prevented an access to the website (server listening on port 8888 which is blocked by a firewall)

		expect a non-bioinformatician to purposfully select between DESeq and DEseq2 ? So either make a website where a lot of choices are already fixed bij a knowledgable bioinformatician (or set a default) so the non-bioinformatician can use it easily (but with limited choices), or make a website for bioinformaticians whith a lot of choices in methods to select from, to ease the burden of parsing all types of data. And then you're looking at something like Galaxy."	
D	More types of data	"A powerful data input manager in case my data is not in a "standard format". Also, I would like the system to be generalized and able to work with DNA data as well."	-

Table 16 Comments on the usability of the current web interface or on core of the web app

As can be seen from Table 16, the main issue arise by what appear to be an unbalanced website (respondent C was a bio-informatician) for biologists, it is a very interesting remark that hits the core of our research question. What kind of information is suitable for biologists to reproduce/reuse studies or simply analyze data? Showing a list of available R packages with their explanations might not be the solution. Besides, the amount of exploratory data analysis elements should be increased. This will undoubtedly have an impact on the quantity of resources produced, stored or rendered.

7.2.4 Threats to validity

The construct validity of the questionnaire is not optimal. One question about the level of difficulty of 4 KDD steps has been dropped as respondents were expected to grade four steps on a scale from 1 (difficult) to 4 (easy). As an example, for some records there were two times the same value entered for a question in contradiction with the statement asking to order all responses with non-unique values. This threat is somewhat limited in our case as we do not draw strong conclusion from these questions but rather describe how the responses are distributed on independent ordinal scales. These scales are not aggregated into constructs. This is different for the SUS-Scale for which the internal validity has been assessed by previous studies (Bangor, Kortum, & Miller, 2009b). One entry for the SUS-Scale has been dropped as the respondent clearly stated to have been unable to judge the tool properly due to its specificity.

The external validity is mainly threatened due to the low sample size. Nevertheless the sample is quite representative of lab or computer people. We expect to find similar trends wherever biology and bioinformatics are involved.

7.3 Wrap-up

First, we see a strong separation between the two types of stakeholder. Biologists are searching for what can be described as these integrative solutions with a lot of visualization, which is indeed close to the HCI-KDD process (Holzinger, 2013). But from the perspective of bioinformaticians, this process is not well adapted as they strongly reject "user-friendly interfaces" and prefer scripts.

Bioinformaticians are also not incline to leave a full KDD process open to biologists as they would not understand what is happening inside the methods or produce awkward data analytics.

Results of the focus groups and which were also confirmed by our survey are summarized in Table 17.

ASPECT	DESCRIPTION	SUPPORT	IMPLEMENTED	MOSCOW
METHOD RECOMMENDATION	Based on uploaded data and knowledge from experts, methods are available as default with their parameters. Also suggestions can be made per type of analysis based on the goal.	3	Static classification	M
EDA OR VISUALIZATION	Offer more visualization to end-users (here biologists). For intermediate transformations and results of analytics	3	Dynamic plots for outlier results and data set comparison	M
CONTEXT INTEGRATION	Adding explicitly what the hypotheses and context are. Also for the methods used while keeping it as much “linked” to external data source that are up-to-date	3	Partial with virtual experiments mockups. Explanations of method provided as links to documentation	M
WORKFLOWS	Editable workflows on the fly. Starting with a prebuild workflow for an experiment, scenarios can be created by editing one or more component	2	No	S
SEARCH	Searching inside the data sets for specific samples or features	2	Filtering per features and automatic comparison on id available but no extensive search.	S
CUSTOM-PACKAGE	Instead of using R packages that are too complex or technical they are replaced by a special custom package integrating these methods (as a workflow for instance).	2	Outlier analysis is actually an example of a custom-package	S
MERGE	Merging data sets is challenging and is handy if processed by such a tool	2	No	S

DATA QUALITY CHECKS	The whole pipeline is represented and lower level files give insights on the quality of the data (reads, mapping…)	2	No	S
COMPARISONS	The influence of the methods are comparable. For instance a double plot showing what happens with one method and the other one next to each other	2	With visualization (double plots)	S
CLUSTER (HPC)	Suggest an explicit binding to a high performance compute facility and let end-users create jobs (and eventually manage them)	2	Mockup of job creation as no real cluster is used for the prototype	C
DYNAMIC DOCUMENTS	Exporting an analysis as a dynamic document	2	Static example of IPython document	C
REPRODUCIBILITY SCORE	Give insights on the similarity of the results between two methods based on machine learning techniques	1	No	W

Table 17 key aspects and their support. Maximum support is 3 (organized focus groups)

Second, as will be discussed again in the concluding chapters, the results of the internal and external focus groups are biased towards bioinformaticians. From our exploratory focus groups and survey, where more biologists were present, there is no indication that the situation is as dramatic as it sounds from the last two focus groups results. A bioinformatician present in the exploratory focus groups judged beneficial to enable KDD as it would allow him to work on his analyses without having to process requests from biologists all the time. In other words, there is some space to let them analyze their data in a sense close to self-service business intelligence, i.e. enable non-experts to retrieve information from data sets (Abelló et al., 2013).

Simply put, it is a major threat to the adoption of such tools by bioinformaticians, focused on the end-user interface and not the API present which was nevertheless considered as a positive component of the architecture to retrieve data sets via scripts. Still, as our overarching research problem is anchored in reproducible research and computational experiments with lots of different intervening actors a balance has to be found and is presented in Chapter 8.

Chapter 8 Design proposition RRO-KDD

In this chapter, both the designed architecture and the evaluation are merged to create a general knowledge discovery process which would be useful and combining resources with enough interactivity. We have seen that reproducible research is satisfied with code and data (Peng, 2011), as it emerges from a purely computational perspective. But we have also seen that actors participating in a "computational experiment" are diverse. That methods or package are not as easily understood even with a link to documentation, as raised by our survey and focus groups. Biology is data intensive, as it should be clear from now on, biologists want tools to visualize and mine data sets (the starter for our own project) and also create scripts (Loman & Watson, 2013) to tackle their data deluge. How to integrate this in a "reproducible" process is suggested in this chapter.

In this chapter we build upon the standard KDD process described in Chapter 5 and the HCI-KDD process suggested by Holzinger (2013). The design proposition therefore takes into account the broader scope of the standard KDD process while being critical at the HCI-KDD and how to involve end-users in the whole process by elaborating on the feedback received during the focus groups and the overall design of our prototype.

The RRO-KDD is modeled with a process-deliverable diagram, a technique elaborated by Weerd & Brinkkemper (2008). First, an overview of the process is given. Next, details about the steps and how to integrate sufficient information are suggested accompanied by sub-PDDs.

8.1 Process-deliverable diagram

Figure 61 depicts the high level KDD process. This process is an outcome of the reflection phase of our design science research.

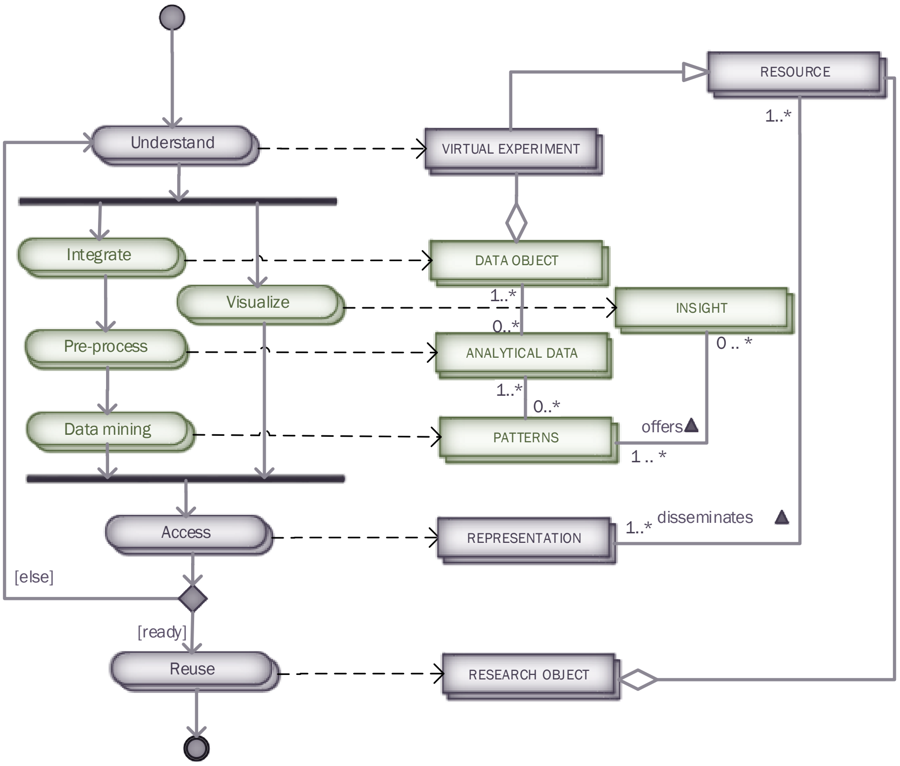


Figure 61 Big picture of the RRO-KDD process

8.2 Description of Concepts

The main concepts of Figure 61 are listed in Table 18. A short description is given for each concept. These are the main deliverables that are expected from the contextually enriched knowledge discovery process we identified as RRO-KDD.

CONCEPT	DESCRIPTION
VIRTUAL EXPERIMENT	Display a set of data sets, methods/packages to an end-user. It can be seen as the human version of a research object.
DATA OBJECT	A digital object referring to a data set.
INSIGHT	Understanding phenomena through visualization (Saraiya, North, & Duca, 2004)
ANALYTICAL DATA	The pre-processed data on which the analysis/data mining is done.
PATTERN	Model describing part of the data
RESOURCE	Any component of a virtual experiment which is dereferenceable and possesses a unique, public identifier.
RESEARCH OBJECT	Aggregates resources and use ontologies to describe resources. It can be seen as the machine readable version of a virtual experiment.

Table 18 Concept table

8.3 Elaborating on open activities

Here we provide explanations at a slightly lower level about the activities and deliverables involved. We deliberately took some liberties with the modelization as the main goal is to make it easily readable. The current state of the RRO-KDD is still a design proposition that we hope to refine later, as explained in future work (section 9.3).

8.3.1 Understand

We encapsulate the context of a computational experiment with sufficient information about the context. As sufficient information there is extracted from MIAME standard or research objects.

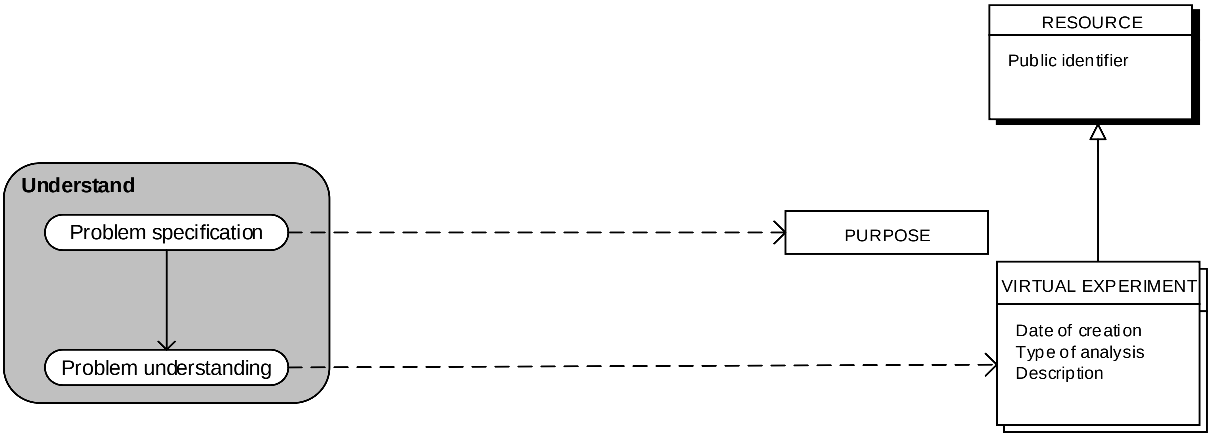


Figure 62 PDD of activity understand

Figure 62 combines the activities of problem specification and understanding as this is collected in a virtual experiment. On screenshots of the prototype, it is the first column to appear when starting an outlier mining experiment.

8.3.2 Integrate

To find an example of data integration, there is no need to go as far as integration of unstructured data or imagery material although it is a scenario that occurs (Holzinger et al., 2014). Integration may be a matter of low support or rare occurring events that requires different data sets that are generated internally or externally. Typically, in our case, finding fusion genes in colorectal tumor samples. Another requirement for integration is validation of findings on one data set. Public data sets can be used to apply an internally developed method like classification of samples in different prognosis categories (De Sousa E Melo et al., 2013).

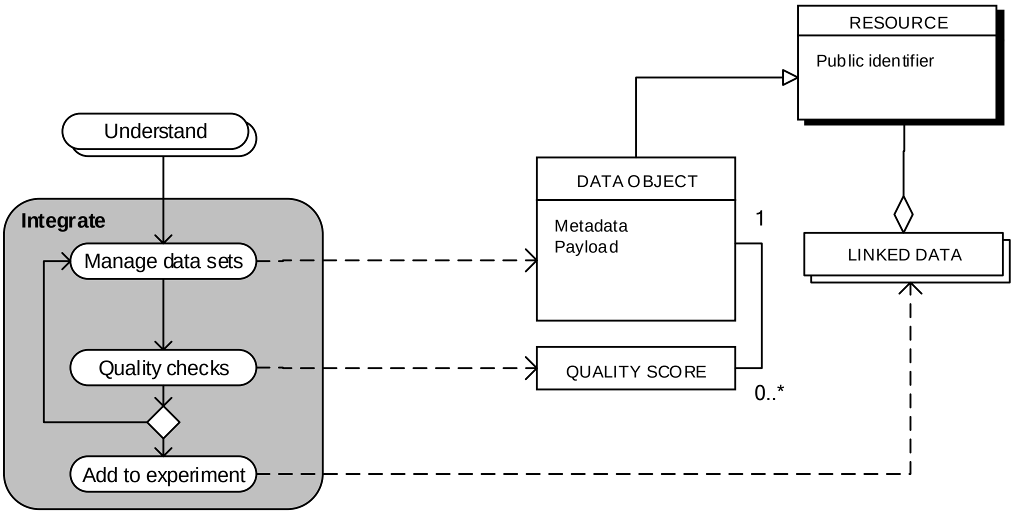


Figure 63 PDD of activity *Integrate*

Figure 63 uses data sets as FAIR data objects. It implicitly requires that such a system is available as the contextual tracking and additional resources are linked to data sets that are themselves organized according to the web architecture.

8.3.3 Pre-process

The pre-processing steps are extracted from García et al. (2015) as they are suitable for our generic model of pre-processing illustrated in Figure 64. After the integration of data sets a crucial, as mentioned before, step requires many different methods. A constraint in our architecture is that a method is uniquely identified. This leads to maintenance issues of keeping these methods up-to-date and a strong selection of the methods available in the tool as it is unrealistic to store every existing method.

After pre-processing, the newly created data set is stored as a data object (with identifier and metadata). Next, it can be accessed like the raw data imported during the data set selection step during integration.

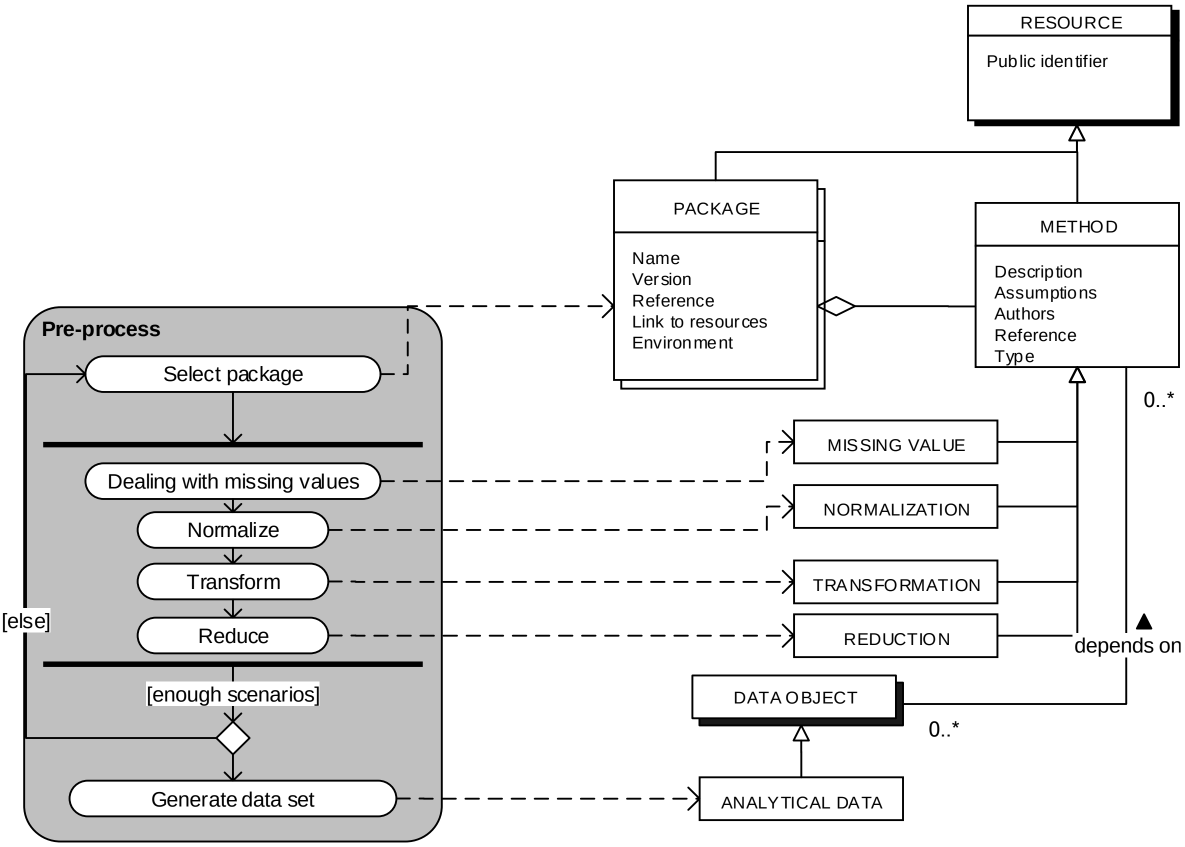


Figure 64 PDD of activity pre-process

Figure 64 illustrates how different modifications affects the original data source. As we have seen previously, our main data sets had no missing values… This is rarely the case and tracing wich elements are removed or what strategy is selected to deal with missing data appears to be more complex. Embedding these decisions in custom packages might facilitate the work of biologists with an although the notions of filtering/removing subsets of the data sets has been raised in our Focus groups by non bioinformaticians.

8.3.4 Data mining

For data mining we implement under the concept "template" an interface to enter parameters before running a package. A template is prepared for data mining with default parameters or a restricted choices of parameters. The parameters entered are recorded as a data mining task. It has to be *reusable*. Execution is done via the concept of tasks and ends as a result, like a list of outliers but it could be any output (classification tree…). The output depends on the package selected based on the purpose. It was recommended during the focus groups to focus less on the techniques and more on the goal a researcher wants to reach. This has been adapted in the activity as selecting a package by purpose. Figure 65 illustrates this adaptations with a high-level overview of what data mining is in our case.

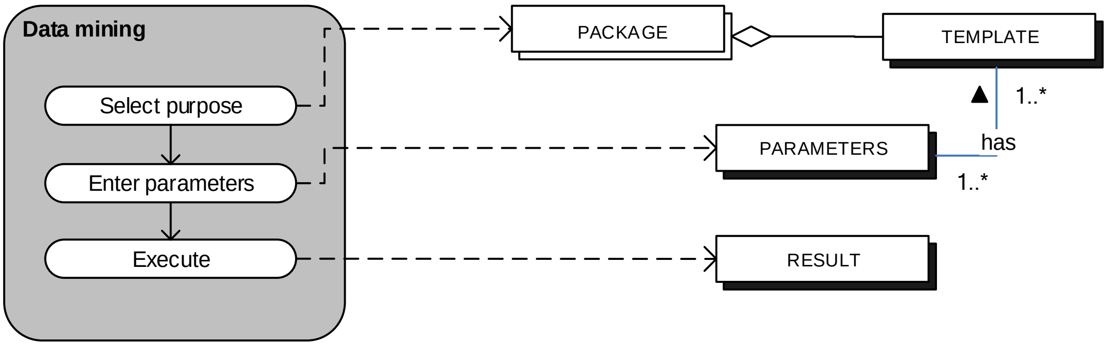


Figure 65 PDD of activity data mining

It is difficult to escape the fact that parameters must be recorded for reusability. These parameters defined in a template may automatically appear on a website with the frameworks we used (Angular.Js). Templates represent the filtering that is asked to be implemented between packages and the end-user.

8.3.5 Visualize

The visualization step is done in parallel, all the steps of KDD being visible to end-users it is recommended to embed visualization for all of them. Despite visualization is seen as the last step of the KDD process in general (Fayyad et al., 1996), evidence collected from focus groups with biologists indicate a strong need to integrate it at each stage. More visualization complies with the recommendations of Holzinger (2013).

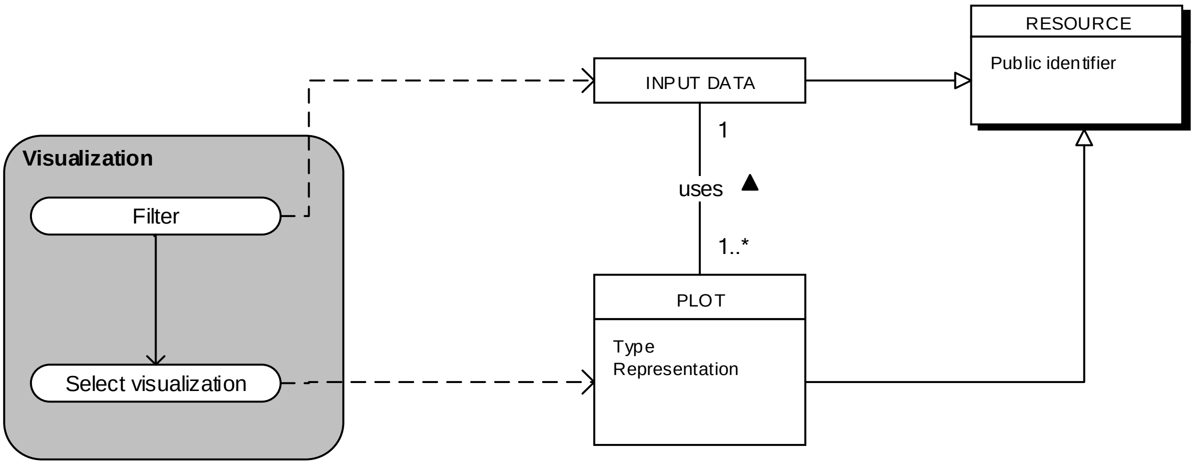


Figure 66 PDD of activity Visualize

Figure 66 shows that filtering capabilities need to be present. Again we observe that a plot is also a resource and has therefore a public identifier to be consulted after its generation.

8.3.6 Access

Access consists in (1) querying a resource, according to the REST semantics described earlier and (2) execution of tasks. A task will apply a method (package) to a data set. As it may be computationally intensive, remote execution on clusters or other powerful IT systems are mandatory. A Task deals with the code execution server-side, not on the client side. A major issue would be an *external* access to reproduce a study for instance. Suggestions to counter this is the distribution of packages as dynamic documents, virtual machines or containers. In that case, the client has to execute the packages on his own infrastructure.

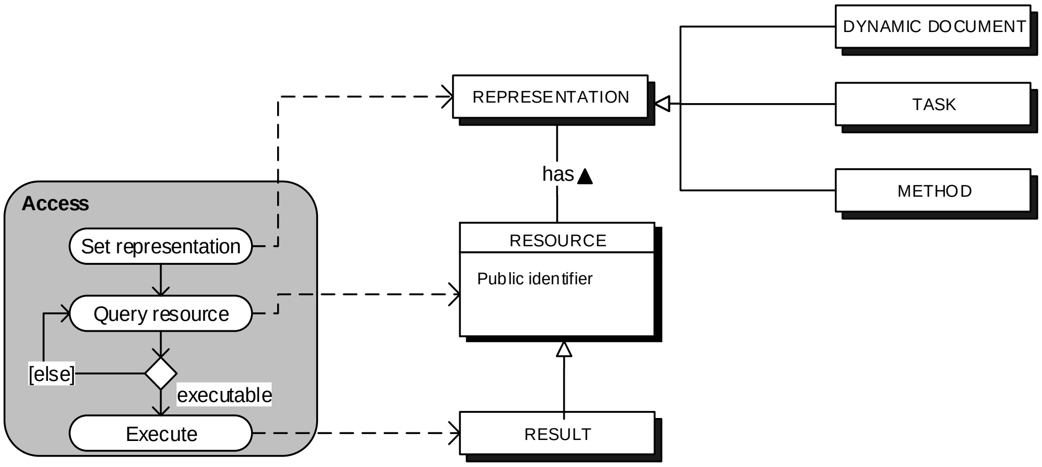


Figure 67 PDD of activity Access

Figure 67 gives an overview about queryable resources. Because of their abstraction level, they fill the role of transformers defined in compendiums and are therefore capable of executing programs remotely or provide dynamic documents, Json results etc.

8.3.7 Reuse

To reuse components of the experiment a sort of experiment repository is needed. A suggested entity to store or disseminate these experiments is named *Research object.* The primary goal of ROs are to be reused, cited and shared (Bechhofer et al., 2010). Reproducibility is one of the goals of ROs which are therefore required to be executable. But at that stage of the experiment, the “tool” produced resources that allow to execute code or visualize dynamic plots. Hence, our research objects are simply semantically enriched virtual experiments (VE) as VEs are more at the user side and decouple the management of virtual experiments from their presentation as Research objects.

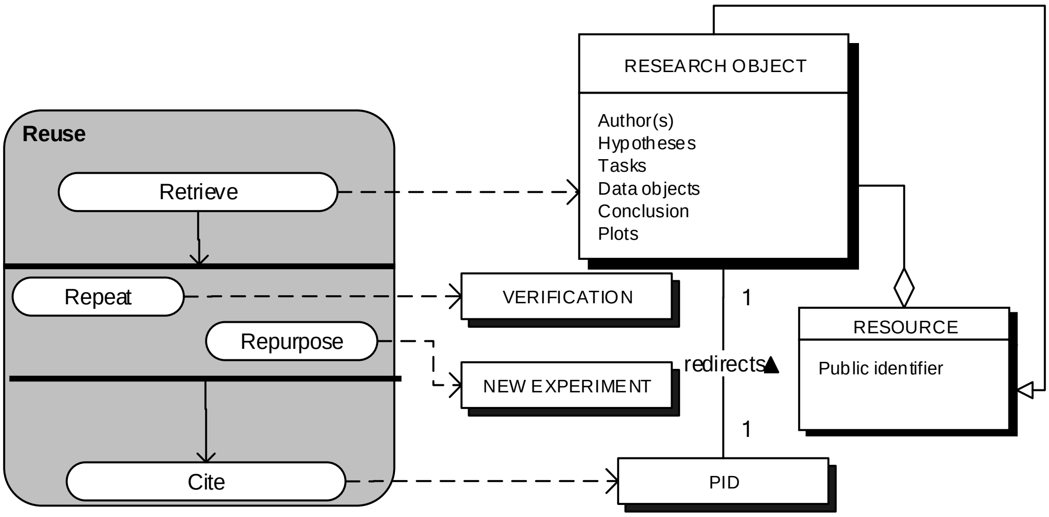


Figure 68 PDD of activity Reuse

Figure 68 illustrates this reuse of a previous experiment available as a research object. By doing this, we clearly combined compendiums and research objects design and enable compendiums on the web. As seen as an important aspect is authorship. Here, we implement PIDs that are redirecting toward a research object to suggest their availability through time.

8.4 Description of Activities

Here we describe the main activities of the RRO-KDD. They are illustrated with examples from the prototype we developed. In the current state, the RRO-KDD is still too much related to genomic data analysis or file formats. We may argue that the main activities are quite generalizable but the sub-activities are not.

ACTIVITY	SUB-ACTIVITY	DESCRIPTION
UNDERSTAND	Problem specification	What is the purpose of analyzing the data at hand
	Problem understanding	What is recorded, what the data represent, how it was generated etc.
INTEGRATE	Manage data set	Integration is used in a very broad sense here, data set selection might be done via external providers or uploading a local data set via the web tool.
	Quality checks	If BAM or reads are uploaded, at the start of the pipeline, scores contained in these files serve to assess the quality of the data. For files like count, a possibility to investigate samples that should be excluded from the data would be part of a quality check.
	Add to experiment	If the data set is retained it can be linked to an experiment and become part of an analysis.
PRE-PROCESS	Normalize	Transformation of raw attributes to analytical variables (García et al., 2015)
	Transform	Creation of new attributes (like linear transformation) (García et al., 2015)
	Deal with missing values	There are several strategies and algorithms that can be applied on missing values. Or sometimes arbitrary

		decisions to drop one or more records that have missing values
	Reduce	Dimensionality reduction (e.g. PCA…)
DATA MINING	Select purpose	As package are categorized by purpose rather than the method they implement a selection based on the goal of the analysis
	Enter parameters	The end-user enters parameters that are recorded next.
	Execute	When executing a DM task, additional computational resources might be required.
ACCESS	Set representation	Before querying a resource an output representation can be chosen. For instance, accessing a chart might return a dynamic plot or a PDF on demand.
	Query resource	Using the HTPP verbs related to semantics specific to REST, resources can be queried, created, updated or deleted
	Execute Task	An existing task can be run again with the same parameters via the same mechanism, the difference is that a task will apply a method/package on a data set.
REUSE	Repurpose	RO as continuity for alternative analyses
	Cite	The ultimate objectives of RO is to be Referenceable from papers, for instance, with an identifier.
	Retrieve	ROs are designed to shareable among the scientific community and other interested third parties.
	Repeat	Execute the study again with information contained in the RO
VISUALIZE	Filter	Before visualizing a plot a subset of the data can be selected from a grid.
	Select visualization	Different plots are suggested to help researchers get insight from their data

Table 19 Activity table

Chapter 9 Conclusion

In this chapter we summarize the results of our design science research and evaluation. One development cycle has been performed and evaluated by three focus groups and one survey.

9.1 Conclusion

This section reiterates the sub research questions and ends with an answer to the main research question of this thesis.

SQ1.1:"What is reproducible research for computational experiments, how is it defined and what existing methods and tools support it and how?"

Reproducible Research defends that the notion of *reproducibility* (or replication) is the *key* of the scientific method. It might raise suspicion on published material without code or data available, as shown by "counting attached files" studies published by RR advocates. Regarding code, parts of a computational experiment is not done purely with scripts and there are still many unsolved technical issues to share them. Certainly if it needs to be executable. As such, RR appears like an elegant driver to design better systems but should not be reduced to a rule of thumb like the gold standard of Peng (2011) shown in section 3.1 . Results from our study suggest that data and code availability to get identical results is not of primary interest for researchers, at least for our respondents, as reusing parts of previous work.

We also noted a weak trend to verify previous work. Even *partial replication* was not a strong demand (see survey results). It is still seen as a "could be" feature. Making RR work, as it would be a useful according to participants, causes the overall knowledge discovery process to become substantially more complex with the tracing of contextual aspects.

SQ1.2:"What aspects are relevant to communicate along with results of an analysis and how to make a computational experiment accessible to the research community?"

While RR is satisfied with code and data sets. Other authors claim that context (protocols, methods, models) are also shareable entities that help third parties to understand or verify previous work. It puts computational aspect as one part of the study that is not in any case sufficient. A better accessibility to computational experiments requires keeping information about scripts developed in bioinformatics but also processing or visualization done by biologists.

SQ1.3:"What are the data (pre)processing and visualization techniques that are relevant to interactively explore gene expression RNA-Seq data and visualizing outliers?"

Techniques will not be summarized here as they are explained in Chapter 4. What can be said is that at the pre-processing stage, which is a determinant step for the rest of the analysis, models are nearly as abundant as data. It partially explains why there is such an analysis bottleneck, more than storing and managing terabytes of data. Diversity seems a criterion that must be wisely considered as seeking to (over)standardize threatens newly published packages or formats.

Besides, only four Bioconductor packages were implemented for RNA-Seq data pre-processing. A research on RNA-Seq packages in Bioconductor yields 1746 entries at time of writing. Additions of packages are frequent because of adaptations of models or new packages to… reproduce previous studies.

SQ1.4:"What are the functional and technical components of a web application that enable or limit reproducible data (pre)processing, interpretation and exchange?"

From what we have seen, a piece of software might collect information about methods and enable an easier code execution, even in different environments. But *reproducibility* has to be (or stay) a quality attribute rather than the goal. As such, what limits this “reproducibility” is that a product will not be used if it does not help to tackle serious data management issues and visualization (the HCI part). It’s the primary interest of people involved in this study to get insights from data, reproducibility is not a goal or a requirement per se.

Technically speaking, such systems become as demanding as professional, general public, applications where design, responsiveness and linkage to external sources come first. From a developer side, these applications should not be as hard to build to fasten their deployment for researchers. We found REST as a well-known and widely used architectural style capable of supporting the most advanced ideas of reusable components, like compendiums.

SQ1.5: “What are the characteristics that have to be implemented in a knowledge discovery process to stress reproducible data (pre)processing, interpretation and exchange with the research community?”

Based on the web architecture and Research objects, an intensive usage of resources that may render plots or execute code remotely and deliver results is suggested as part of the implementation of KDD steps in a mining tool. This to connect the application with ongoing research on exchangeable components of computational experiments.

To conclude, we answer our main research question:

> *MQ: “How to ensure that a computational experiment conducted by means of an interactive knowledge discovery process using web resources and technologies offers an adequate level of reproducibility?”*

As has been explained earlier “adequate level” refers to the sufficient amount of information necessary to *reproduce* a study. Reproduction does not necessary covers all the aspects of a study, e.g. an entire analysis pipeline and the experiments in a lab. *Replication* requires contextual, data, code and algorithmic information even for a *partial reproducibility* scenario.

We suggested an aggregation of resources in research objects that would be adequate for *partial* reproducibility and hence reusability. The challenge is if we target full reproducibility. As far as our prototype and evaluation suggest, there is no acceptable solution that *ensures full reproducibility of a computational experiment*. Reasons are described earlier in the answers to sub research questions. In case of *partial reproducibility,* the availability of components of an experiment and its context might, if not ensure in all cases, enable *reusability* better than solutions discussed in the literature of Reproducible Research.

As we may conclude from our qualitative evaluation, *reproducibility* is not perceived as important as reusability. It means that a “reusable” implementation of an algorithm with other parameters and other data is worth some efforts to implement. In addition, it is expected to be reusable in different contexts like scripting in bioinformatics or visualization in biology. Letting third-parties rerunning code as-is for verification or understanding appears to be negligible from a research perspective as the disinterest in such practices are prominent, according to our evaluation results.

9.2 Discussion and limitations

A strong debate emerging from the focus groups was if we let biologists perform the steps that fall under the KDD part or if we leave only HCI in an application with data sets that are ready-to-use. As said, the HCI-KDD models a single user performing every tasks via interactive visualization solutions whereas this does not hold in our settings. Still, we recommend here to follow the merge between HCI and KDD as the demand is strong for that. Despite that fact, we have to calibrate the low-level methods with “custom-packages” that perform the required steps and are in other words “biologist-friendly”. What is important here is to give access to details about the low-level models but not necessarily leave as much freedom for their selection as we did.

Adaptations of this view are therefore welcome as different stakeholders are involved in the process. All of them don’t expect the same from an interactive interface or don’t expect anything interactive. Moreover, from a bioinformatics side KDD and HCI should be separated while for biology there is indeed a call for a merge between both aspects. This problem of accessing information is quite close to the “1 has access but 99 need to have access” (Weber, 2013) that seems prevalent in the business intelligence domain.

A single design science study does not explain it all. Despite that problems with *omics* data are shared among many research labs in the world, the non-standardized analytics applied to it makes a tool such as the prototype developed too specific. Many different kind of file formats and methods are not explored in this research. The same holds for more complex experimental designs. Decisions such as strategies to deal with missing data and how to properly track and contextualize these decisions are not investigated. , There are many unsolved reproducibility issues partially due to the inapplicability of solutions described in Chapter 3 in real settings.

Also, a research object mixes elements from lab research and computational parts of an experiment. The nature of contextual information related to lab experiments or potentially extracted from modules or packages for data analytics makes it hard to provide clear guidelines. Without such guidelines, we are lacking a more precise model of sufficient information to store and exchange ROs. For now, *minimum information* standards are considered but their application in practice has not been evaluated with the prototype.

Another limitation is that the *machine readable* side of data exchange has not been implemented and evaluated. From start, we are restricted to what users want to interact with. ROs are semantically enriched messages with custom ontologies extending OAI-ORE. The suitability, feasibility, effectiveness and efficiency of deploying an architecture based on these messages is not examined.

9.3 Future work

We uniquely combined the notions of interactivity and Research Objects. A lot of research lays ahead to provide useful reusable components of a computational experiment. Putting too much emphasize on describing or extending ontologies may slow down the technical implementation of such repositories and knowledge dissemination. Nevertheless, room for improvements for sharing components and enable partial reproducibility are there. Solving other challenges as custom-package construction and usage by biologists in practice is primordial.

For Reproducible Research, we believe that more investigation is needed with practitioners involved and that counting code or data sets available with papers does not clarify what is happening in these fields. There are still non-elucidated challenges of combining KDD and HCI. How to mine data sets interactively while generating *sufficient information* is a goal still unreached.

One supplementary bottleneck for more interactive solutions, is the relation between bioinformatics and biology. As we have seen, there are demands for easier data analytics solutions. Self-service data analytics in these fields seems unavoidable. It would nonetheless require serious investigations on the best ways to design analysis or mining solutions further for these users (i.e. also outside research) as interpretation was a key issue raised by our Focus groups.

Regarding Research objects, most of the literature is focused on workflow-centric ROs. Other types of research objects are also suggested in the literature. Called working objects, view objects or method objects etc. (Bechhofer et al., 2010). These types of objects are thought to fit in an overall "research" object management architecture. We may argue that while further research is mandatory to make ROs fit the needs of exchanging resources, increasing subtypes of objects appears like over-engineering. A simplification of these objects would be appropriate. Indeed, we have seen that *reproducibility* is not a main concern but *reusability* is seen as a major improvement. A reflection has to be conducted on how to make ROs management more direct and transparent as end-users emphasize data management and visualization issues first. It is doubtful if RO complexification, by adding subtypes or specific ontologies, will have the intended benefits in real settings.

Next we comment on the outputs from our evaluation in Table 17, Chapter 7. Each element is discussed, except cluster and reproducibility score, based on what was implemented and what is still to be investigated.

First, method recommendation. In the prototype a static classification was offered (see Table 8) to users to assist package selection. This classification is done by reading through the instructions and manually judging where they fit while a more automated approach might make this classification easier. The problem is similar to model *curation* which is also done manually (see BioModels). More research on how existing recommendations system might be applicable to packages or algorithms dynamically called by a tool would be beneficial.

Second, exploratory data analysis and visualization with custom-packages, data quality checks and comparison of data set is a challenge. These elements are combined as the goal is to explore data based on "pre-made" packages that would facilitate the analysis. It would be worth to investigate how optimal custom packages can be developed and integrated. Some respondents suggested a workflow edition of the main components of a pipeline. This makes sense as quality checks, Phred scores for instance, can be encoded differently according to the low-level analysis tools used on DNA sequence files and sequencing platforms. More guidelines on a skeleton for custom packages are worth to be studied. In addition, exploring to what extend workflow edition in galaxy can be adapted to the expectations of a visualization interface may facilitate the implementation. This framework could serve as a pillar for tracking and package management.

Next, context integration covered *search* and *dynamic documents*. Future work is necessary on how to best implement search tools. They have to retrieve up-to-date information about methods but might also search on the meta-data. This seems to be very technology-dependent. In our case an abstraction layer, independent of the type of data base or web service used, must be able to render this information to users. For dynamic documents, automated generation of code executed server-side and information from virtual experiments and methods could improve communication with other teams. But how and to what extend is still unknown.

Then, the RRO-KDD process attempts to reunite contextual information already included in previous KDD processes with the hype of data-driven processes like HCI-KDD. The latter is more centered on algorithms and visualization. The RRO-KDD process needs to be evaluated and improved. Or by being fully implemented in a tool or by challenging all the concepts and activities developed in the RRO-KDD by applying them to more data analysis cases.

Last, challenges identified earlier are that improving the architecture and Research Objects design is based on end-user input as the HCI part is embedded in the RRO knowledge discovery process. Working on the HCI part of the process will increase the probability of catching the interests of users. Research objects design for reproducibility reasons might not make them as enthusiastic. Hence, it forces to work on RO design in parallel to visualization and interfaces design.

Ultimately, data management and context management are merged together and components are made easily retrievable. Improving reproducibility or reusability is possible even for highly interactive data mining applications. However, it necessitates a combination of many different fields of research. Exchanging or retrieving contextual data exceeds by far the design of interactive applications. The key is to link all aspects of the RRO-KDD together and not develop them separately. This will guarantee that evolutions are anchored in current practices and technologies while benefiting from the input of stakeholders through concrete interfaces and systems.

References

Abelló, A., Darmont, J., Etcheverry, L., Golfarelli, M., Mazón, J.-N., Naumann, F., … Vossen, G. (2013). Fusion cubes: Towards self-service business intelligence. *International Journal of Data Warehousing and Mining*, *9*(2), 66–88. doi:10.4018/jdwm.2013040104

Adhianto, L., Banerjee, S., Fagan, M., Krentel, M., Marin, G., Mellor-Crummey, J., & Tallent, N. R. (2010). Towards open science: the myExperiment approach. *Concurrency Computat.: Pract. Exper.*, *22*, 2335–2353. doi:10.1002/cpe

Anders, S., & Huber, W. (2010). DESeq: Differential expression analysis for sequence count data. *Genome Biology*, *11*, R106. doi:10.1186/gb-2010-11-10-r106

Anders, S., McCarthy, D. J., Chen, Y., Okoniewski, M., Smyth, G. K., Huber, W., & Robinson, M. D. (2013). Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. *Nature Protocols*, *8*, 1765–86. doi:10.1038/nprot.2013.099

Bangor, A., Kortum, P., & Miller, J. (2009a). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, *4*, 114–123. doi:66.39.39.113

Bangor, A., Kortum, P., & Miller, J. (2009b). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, *4*, 114–123. doi:66.39.39.113

Baumer, B., Cetinkaya-Rundel, M., & Bray, A. (2014). R Markdown: Integrating A Reproducible Analysis Tool into Introductory Statistics. *arXiv Preprint arXiv:1402.1894*. Retrieved from http://arxiv.org/abs/1402.1894

Bechhofer, S., De Roure, D., Gamble, M., Goble, C., & Buchan, I. (2010). Research Objects: Towards Exchange and Reuse of Digital Knowledge. *The Future of the Web for Collaborative Science.* doi:10.1038/npre.2010.4626.1

Belhajjame, K., Zhao, J., Garijo, D., Hettne, K., Palma, R., Corcho, Ó., … Goble, C. (2014). The Research Object Suite of Ontologies: Sharing and Exchanging Research Data and Methods on the Open Web. *arXiv Preprint arXiv: 1401.4307*. Retrieved from http://arxiv.org/abs/1401.4307

Borgman, C. L. (2012). The conundrum of sharing research data. *Journal of the American Society for Information Science and Technology*. doi:10.1002/asi.22634

Brazma, A. (2009). Minimum Information About a Microarray Experiment (MIAME)--successes, failures, challenges. *TheScientificWorldJournal*, *9*, 420–423. doi:10.1100/tsw.2009.57

Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. *Information and Software Technology*, *38*(4), 275–280.

Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability Evaluation in Industry*, *189*, 194. doi:10.1002/hbm.20701

Callahan, S. P., Freire, J., Santos, E., Scheidegger, C. E., Vo, T., & Silva, H. T. (2006). VisTrails : Visualization meets Data Management. In *2006 ACM SIGMOD iInternational Conference on Management of Data* (pp. 745–747). doi:10.1145/1142473.1142574

Carlsson, S. A., Henningson, S., Hrastinski, S., & Keller, C. (2011). Socio-technical IS design science research: developing design theory for IS integration management. *Information Systems and E-Business Management*, *9*, 109–131.

Chin, L., Andersen, J. N., & Futreal, P. A. (2011). Cancer genomics: from discovery science to personalized medicine. *Nature Medicine*, *17*(3), 297–303. doi:10.1038/nm.2323

Cooper, J., Science, C., & Road, P. (2014). A call for virtual experiments : accelerating the scientific process. *Progress in Biophysics and Molecular Biology*, 8–15. doi:10.1016/j.pbiomolbio.2014.10.001

Davison, A., & Mattioni, M. (2014). Sumatra: A Toolkit for Reproducible Research. In *Implementing Reproducible Research* (pp. 1–19).

De Sousa E Melo, F., Wang, X., Jansen, M., Fessler, E., Trinh, A., de Rooij, L. P. M. H., … Vermeulen, L. (2013). Poor-prognosis colon cancer is defined by a molecularly distinct subtype and develops from serrated precursor lesions. *Nature Medicine*, *19*(5), 614–8. doi:10.1038/nm.3174

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., … Gingeras, T. R. (2013). STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics*, *29*(1), 15–21. doi:10.1093/bioinformatics/bts635

Donoho, D. L. (2010). An invitation to reproducible computational research. *Biostatistics (Oxford, England)*, *11*(3), 385–8. doi:10.1093/biostatistics/kxq028

Drummond, D. C. (2009). Replicability is not Reproducibility: Nor is it Good Science, (2005), 2005–2008. Retrieved from http://cogprints.org/7691/

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*. doi:10.1145/240455.240464

Fielding, R. T., & Taylor, R. N. (2000). Principled design of the modern Web architecture. *Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium*. doi:10.1109/ICSE.2000.870431

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development*, *9*, 28–35. doi:10.1177/004057368303900411

Freire, J., Koop, D., Chirigati, F., & Silva, C. (2014). Reproducibility using VisTrails. In *Implementing Reproducible Research* (pp. 1–26).

García, S., Luengo, J., & Herrera, F. (2015). *Data Preprocessing in Data Mining*. Springer.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., … Zhang, J. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, *5*(10), R80. doi:10.1186/gb-2004-5-10-r80

Gentleman, R., & Lang, D. (2007). Statistical analyses and reproducible research. *Journal of Computational and …*, *16*(1), 1–23. Retrieved from http://amstat.tandfonline.com/doi/abs/10.1198/106186007X178663

Goecks, J., Nekrutenko, A., & Taylor, J. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology*, *11*(8), R86. doi:10.1186/gb-2010-11-8-r86

Goguen, J. A., & Linde, C. (1993). Techniques for requirements elicitation. *[1993] Proceedings of the IEEE International Symposium on Requirements Engineering*. doi:10.1109/ISRE.1993.324822

González-Barahona, J. M., & Robles, G. (2011). On the reproducibility of empirical software engineering studies based on data retrieved from development repositories. *Empirical Software Engineering*, *17*(1-2), 75–89. doi:10.1007/s10664-011-9181-9

Hamburg, M., & Collins, F. (2010). The path to personalized medicine. *New England Journal of Medicine*, 301–304.

Hansen, K. D., Brenner, S. E., & Dudoit, S. (2010). Biases in Illumina transcriptome sequencing caused by random hexamer priming. *Nucleic Acids Research*, *38*(12). doi:10.1093/nar/gkq224

Hevner, A., & Chatterjee, S. (2010). *Design research in information systems*. Springer New York.

Hodge, V. J., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*. doi:10.1023/B:AIRE.0000045502.10941.a9

Hoefling, H., & Rossini, A. (2014). Reproducible Research for Large-Scale Data Analysis. In *Implementing Reproducible Research* (pp. 1–17).

Holzinger, A. (2013). Human-Computer Interaction and Knowledge Discovery (HCI-KDD): What is the benefit of bringing those two fields to work together? In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 8127 LNCS, pp. 319–328). doi:10.1007/978-3-642-40511-2_22

Holzinger, A., Dehmer, M., & Jurisica, I. (2014). Knowledge Discovery and interactive Data Mining in Bioinformatics - State-of-the-Art, future challenges and research directions. *BMC Bioinformatics*, *15 Suppl 6*(Suppl 6), I1. doi:10.1186/1471-2105-15-S6-I1

Holzinger, A., & Jurisica, I. (2014). Knowledge Discovery and Data Mining in Biomedical Informatics : The Future Is in Integrative , Interactive. *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, 1–18.

Hothorn, T., & Leisch, F. (2011). Case studies in reproducibility. *Briefings in Bioinformatics*, *12*(3), 288–300. doi:10.1093/bib/bbq084

Ioannidis, J. P. A. (2014). How to Make More Published Research True. *PLoS Medicine*, *11*(10), e1001747. doi:10.1371/journal.pmed.1001747

Jurisica, A. H. (2014). Knowledge Discovery and Data Mining in Biomedical Informatics: The Future Is in Integrative. *Interactive Machine Learning Solutions*, *Interactiv*(NA), 1–18.

Kemper, K., Versloot, M., Cameron, K., Colak, S., De Sousa E Melo, F., De Jong, J. H., … Medema, J. P. (2012). Mutations in the Ras-Raf axis underlie the prognostic value of CD133 in colorectal cancer. *Clinical Cancer Research*, *18*(11), 3132–3141. doi:10.1158/1078-0432.CCR-11-3066

King, G. (1995). Replication, Replication. *PS: Political Science & Politics*, *28*(3), 444–452.

Knuth, D. E. (1984). Literate Programming. *The Computer Journal*, *27*(2), 97–111. doi:10.1093/comjnl/27.2.97

Kuehn, H., Liberzon, A., Reich, M., & Mesirov, J. P. (2008). Using GenePattern for gene expression analysis. *Current Protocols in Bioinformatics / Editoral Board, Andreas D. Baxevanis ... [et Al.]*, *Chapter 7*, Unit 7.12. doi:10.1002/0471250953.bi0712s22

Laine, C., Goodman, S. N., Griswold, M. E., & Sox, H. C. (2007). Reproducible Research : Moving toward Research the Public Can Really Trust. *Annals of Internal Medicine*, *146*(6), 450–453. Retrieved from http://annals.org/article.aspx?articleid=733696

Law, C. W., Chen, Y., Shi, W., & Smyth, G. K. (2014). Voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology*, *15*(2), R29. doi:10.1186/gb-2014-15-2-r29

Leisch, F. (2002). Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis. In *COMPSTAT 2002 Proceedings in Computational Statistics* (pp. 575–580). doi:10.1.1.20.2737

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., … Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics (Oxford, England)*, *25*(16), 2078–9. doi:10.1093/bioinformatics/btp352

Liu, Z., & Pounds, S. (2014). An R package that automatically collects and archives details for reproducible computing. *BMC Bioinformatics*, *15*(1), 138. doi:10.1186/1471-2105-15-138

Loman, N., & Watson, M. (2013). So you want to be a computational biologist? *Nature Biotechnology*, *31*(11), 996–998. doi:10.1038/nbt.2740

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-Seq data with DESeq2. *Genome Biology*, *15*(12), 550. doi:10.1101/002832

McNutt, M. (2014). Journals unite for reproducibility. *Science*, *346*(6210), 679–679.

Menzies, T., & Shepperd, M. (2012). Special issue on repeatable results in software engineering prediction. *Empirical Software Engineering*, *17*(1-2), 1–17. doi:10.1007/s10664-011-9193-5

Mettler, T., Eurich, M., & Winter, R. (2014). On the Use of Experiments in Design Science Research: A Proposition of an Evaluation Framework. *Communications of the AIS*, *34*(1), 223–240.

Mons, B., & Velterop, J. (2009). Nano-publication in the e-science era. In *CEUR Workshop Proceedings* (Vol. 523). doi:10.3233/ISU-2010-0613

Moseley, E. T., Hsu, D. J., Stone, D. J., & Celi, L. A. (2014). Beyond Open Big Data: Addressing Unreliable Research. *Journal of Medical Internet Research*, *16*(11), e259. doi:10.2196/jmir.3871

Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., … Li, P. (2004). Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, *20*, 3045–3054. doi:10.1093/bioinformatics/bth361

Omta, W. A., Egan, D. A., Spruit, M. R., & Brinkkemper, S. (2012). Information Architecture in High Throughput Screening. *Procedia Technology*. doi:10.1016/j.protcy.2012.09.077

Osborne, J. M., Bernabeu, M. O., Bruna, M., Calderhead, B., Cooper, J., Dalchau, N., … Deane, C. (2014). Ten Simple Rules for Effective Computational Research. *PLoS Computational Biology*, *10*. doi:10.1371/journal.pcbi.1003506

Pachter, L. (2012). Models for transcript quantification from RNA-Seq, 11–12. doi:10.1038/nbt.162

Pautasso, C. (2014). REST: Advanced Research Topics and Practical Applications. Retrieved from http://dx.doi.org/10.1007/978-1-4614-9299-3

Peng, R. D. (2011). Reproducible research in computational science. *Science (New York, N.Y.)*, *334*(6060), 1226–7. doi:10.1126/science.1213847

Peng, R. D., Dominici, F., & Zeger, S. L. (2006). Reproducible epidemiologic research. *American Journal of Epidemiology*. doi:10.1093/aje/kwj093

Pérez, F., & Granger, B. E. (2007). IPython: A system for interactive scientific computing. *Computing in Science and Engineering*, *9*, 21–29. doi:10.1109/MCSE.2007.53

Pries-Heje, J., Baskerville, R., & Venable, J. (2008). Strategies for design science research evaluation. In *Proceedings of the 16th European Conference on Information Systems* (pp. 1–12).

Reich, M., Liefeld, T., Gould, J., Lerner, J., Tamayo, P., & Mesirov, J. P. (2006). GenePattern 2.0. *Nature Genetics*, *38*(5), 500–1. doi:10.1038/ng0506-500

Roberts, A., Trapnell, C., Donaghey, J., Rinn, J. L., & Pachter, L. (2011). Improving RNA-Seq expression estimates by correcting for fragment bias. *Genome Biology*, *12*(3), R22. doi:10.1186/gb-2011-12-3-r22

Roberts, R. J., Carneiro, M. O., & Schatz, M. C. (2013). The advantages of SMRT sequencing. *Genome Biology*, *14*(6), 405. doi:10.1186/gb-2013-14-6-405

Robinson, M. D., McCarthy, D. J., & Smyth, G. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139–140.

Robinson, M. D., & Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol*, *11*(3), R25. doi:gb-2010-11-3-r25 [pii]\n10.1186/gb-2010-11-3-r25

Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLoS Computational Biology*, *9*(10), e1003285. doi:10.1371/journal.pcbi.1003285

Saraiya, P., North, C., & Duca, K. (2004). An Evaluation of Microarray Visualization Tools for Biological Insight. *IEEE Symposium on Information Visualization*. doi:10.1109/INFVIS.2004.5

Sarkar, I. N. (2010). Biomedical informatics and translational medicine. *Journal of Translational Medicine*, *8*, 22. doi:10.1186/1479-5876-8-22

Scholz, M. B., Lo, C. C., & Chain, P. S. G. (2012). Next generation sequencing and bioinformatic bottlenecks: The current state of metagenomic data analysis. *Current Opinion in Biotechnology*. doi:10.1016/j.copbio.2011.11.013

Seshagiri, S., Stawiski, E. W., Durinck, S., Modrusan, Z., Storm, E. E., Conboy, C. B., … de Sauvage, F. J. (2012). Recurrent R-spondin fusions in colon cancer. *Nature*, *488*(7413), 660–4. doi:10.1038/nature11282

Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing*, *5*, 13–22.

Shortliffe, E., & Cimino, J. (2014). *Biomedical informatics*. Springer London.

Smyth, G. (2005). limma: Linear Models for Microarray Data. In *Bioinformatics and Computational Biology Solutions Using R and Bioconductor* (pp. 397–420). doi:citeulike-article-id:5722720

Smyth, G. K., Ritchie, M., Thorne, N., Wettenhall, J., & Shi, W. (2013). limma:Linear Models for Microarray Data User's Guide(Now Including RNA-Seq Data Analysis). *R Manual*, 1–123.

Sonnenberg, C., & Brocke, J. vom. (2012). Evaluations in the science of the artificial–reconsidering the build-evaluate pattern in design science research. In *Design Science Research in Information Systems. Advances in Theory and Practice* (pp. 388–397).

Stodden, V. (2009). Enabling reproducible research: licensing for scientific innovation. *International Journal of Communications Law Policy*, *13*(13), 1–25. doi:10.2139/ssrn.1362040

Tremblay, M. C., Hevner, A. R., & Berndt, D. J. (2010). The Use of Focus Groups in Design Science Research. In *Design Research in Information Systems* (pp. 121–143). doi:10.1007/978-1-4419-5653-8_10

Tukey, J. W. (1977). *Exploratory Data Analysis*. *Analysis* (Vol. 2). doi:10.1007/978-1-4419-7976-6

Van Gorp, P., & Mazanek, S. (2011). SHARE: A web portal for creating and sharing executable research papers. In *Procedia Computer Science* (Vol. 4, pp. 589–597). doi:10.1016/j.procs.2011.04.062

Vandewalle, P., Kovacevic, J., & Vetterli, M. (2009). Reproducible research in signal processing. *IEEE Signal Processing Magazine*. doi:10.1109/MSP.2009.932122

Wagner, P., & Srivastava, S. (2012). New paradigms in translational science research in cancer biomarkers. *Translational Research*, *159*(4), 343–353.

Wang, Z., Gerstein, M., & Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews. Genetics*, *10*(1), 57–63. doi:10.1038/nrg2484

Weber, M. (2013). Keys to Sustainable Self-Service Business Intelligence. *Business Intelligence Journal*, *18*, 18–24.

Weerd, I. Van De, & Brinkkemper, S. (2008). Meta-modeling for situational analysis and design methods. *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications*, 38–58.

Xie, Y. (2014). *Dynamic Documents with R and knitr Yihui*. *Journal of Statistical Software* (Vol. 56).

Zhao, J., Gomez-Perez, J. M., Belhajjame, K., Klyne, G., Garcia-Cuesta, E., Garrido, A., … Goble, C. (2012). Why workflows break - Understanding and combating decay in Taverna workflows. In *8th IEEE International Conference on eScience*. Chicago, USA. doi:10.1109/eScience.2012.6404482

Zwiener, I., Frisch, B., & Binder, H. (2014). Transforming RNA-Seq data to improve the performance of prognostic gene signatures. *PLoS ONE*, *9*. doi:10.1371/journal.pone.0085150

Appendices

A. Focus groups questions

> *"How to ensure that a computational experiment conducted by means of an interactive data mining process offers an adequate level of reproducibility?"*

1) Does the three activities load experiment (context) (1), normalize and clean the data set (2) and select an algorithm (3) make sense and are generalizable to other kind of studies?
2) Are the following components: method (1), package (2), data set (3), chart (4) and job (5) (algorithm & parameters) sufficient for a user to interact with data, why (not)?
3) Context and computational components are both necessary or is the latter enough (context should not be part of data management and analytics, that's the role of papers)?
4) Are technical (implementation) details of a method/package of interest for a biologist? Does this help bioinformaticians?
5) When it is a custom algorithm implemented in the tool, is it judicious to give access to the code via a dynamic document (e.g. IPython notebook), does it have to be easily executable?

B. Survey questions

I define myself as a: a biologist, a bioinformatician

What is your research domain?

Type of your organization

My role in the organization is

On a scale from 1 to 5, how do you judge your computer skills?

1) Basic manipulation of Microsoft Office (or Open Office, Libre Office…), web browser to access data
2) I can switch to the command line if needed (Powershell, bash…) to execute tools that do not have a User-Friendly Interface
3) I know some programming or scripting languages and write small programs myself. I can send jobs to a cluster (or generally communicate via SSH to a remote server)
4) I know some best practices to store and share code (version control). I can work with libraries or modules for advanced data analytics. I can manipulate web services or web APIs
5) I can adapt to different programming paradigms (declarative, imperative, event-driven…) and I am aware of the trends and evolutions in the field of data analysis and databases (i.e. I have the big picture about Hadoop, In memory databases, NoSQL…). I am able to evaluate whether these solutions are appropriate or not to tackle a given problem

1) I feel the need to have more information about a published study than just the paper, supplementary table and supplementary data.

2) I feel the need to repeat a published study entirely (Full replication) to verify the outcomes

3) I feel the need to repeat a published study entirely (Full replication) to understand the outcomes

4) I feel the need to repeat some parts of a published study (Partial replication) to verify the outcomes

5) I feel the need to repeat some parts of a published study (Partial replication) to understand the outcomes

6) It is hard to figure out how the data at hand was generated (e.g. flow from sequencing to summarized information, intervening algorithms) when it is a downloaded data set from a published paper.

7) It is hard to figure out how the data at hand was generated (e.g. flow from sequencing to summarized information, intervening algorithms) when it was created internally (own lab).

8) When a package (e.g. a Bioconductor R package) is provided with a paper, this enables investigation of the computational aspects of the study in a convenient way

9) A classification of packages by methods they implement is easier than by package name and version

10) Package name and version are not enough.

11) The availability of code and data is not enough

12) I would consider to delay the submission of my paper to make sure that my computational experiments (code, data) are executable by an external reviewer

13) Peer-review is sufficient to guarantee the quality of the papers published.

14) Additional material attached to an online paper fulfills all the prerequisites to figure out the computations or methods applied in a paper

15) I experience difficulties to analyze data due to package (or modules) version mismatches when using external code

16) Indicate which languages you like to use to perform data analytics tasks

17) Indicate if you routinely perform the following tasks with you code?

18) I use at least once per month [Galaxy]

I use at least once per month [GenePattern]

I use at least once per month [Mobyle]

I use at least once per month [MyExperiment.org]

19) I consider a package or tool like a black box which is working fine

20) Rank the following tasks that are the most problematic to you and your team when dealing with data. [Understanding the data]
Rank the following tasks that are the most problematic to you and your team when dealing with data. [Pre-processing the data]

Rank the following tasks that are the most problematic to you and your team when dealing with data. [Analyzing the data]

Rank the following tasks that are the most problematic to you and your team when dealing with data. [Visualizing the data]

The following question are from the SUS Scale (Brooke, 1996):

21) I think that I would like to use this system frequently

22) I found the system unnecessarily complex

23) I thought the system was easy to use

24) I think that I would need the support of a technical person to be able to use this system

25) I found the various functions in this system were well integrated

26) I thought there was too much inconsistency in this system

27) I would imagine that most people would learn to use this system very quickly

28) I found the system very cumbersome to use

29) I felt very confident using the system

30) I needed to learn a lot of things before I could get going with this system.

31) This tool makes it easy for biologists to perform basic data mining tasks and help bioinformaticians to get some insight about what kind of analysis were done. What do you believe is really missing?

Universiteit Utrecht

C. Survey detailed results

Table 20 Detailed results

Group	Bioinformatician	Biologist	Bioinformatician	Bioinformatician
Domain	Breast Cancer	Developmental biology	Developmental biology	Cardiovascular
Role	Bioinformatician	Student	PhD	Post-doc
Computer skills	4	1	5	5
Q1	2	4	2	4
Q2	1	2	1	2
Q3	1	2	1	2
Q4	4	3	1	5
Q5	1	3	1	3
Q6	2	3	1	3
Q7	1	3	1	3
Q8	2	3	4	4
Q9	5	3	4	4
Q10	1	4	3	3
Q11	3	4	1	4
Q12	1	4	3	4
Q13	2	5	4	2
Q14	2	4	4	3
Q15	2	3	3	3
Language	Perl, Other	Java/Groovy	R, Python, Bash, Other	R, C
Other	Stata		Visual Basic	
Task			Monitor versions	Save an environment
Galaxy	I don't use it	I don't know what it	I don't use it	I don't use it
GenePattern	I don't use it	I don't know what it	I don't use it	I don't use it
Mobyle	I don't use it	I don't know what it	I don't use it	I don't know what it
MyExperiment	I don't use it	I don't know what it	I don't use it	I don't use it
Q19	3	3	1	3
	R1	**R2**	**R3**	**R4**

Biologist	Bioinformatician	Biologist	Bioinformatician	Biologist
Cancer genomics	Cardiology	Cancer genomics, Cell	Developmental biology	Cancer genomics, Cell
Post-doc	PhD	Technician	Technician	Principal Investigator
3	3	2	4	3
3	4	2	5	4
3	2	1	2	3
2	1	2	2	2
4	3	2	3	4
2	1	2	4	4
4	5	3	4	3
4	3	2	3	2
3	5	2	4	4
3	2	3	3	5
4	4	3	4	5
4	5	3	3	5
4	2	3	4	2
2	1	2	3	4
1	2	2	3	3
3	3	4	3	2
Python	R, Python, Bash	R, Python, Perl	R, Python, Perl, Bash,	R, Python, Perl, Bash
			D	
Generate dynamic	Generate static		Generate static	
I don't use it	I don't know what it	I don't know what it	I don't use it	I don't use it
I don't use it	I don't know what it	I don't know what it	I don't use it	I don't know what it
I don't use it	I don't know what it	I don't know what it	I don't use it	I don't know what it
I don't use it	I don't know what it	I don't know what it	I don't use it	I don't know what it
4	3	4	2	1
R9	**R8**	**R7**	**R6**	**R5**

Bioinformatician	Biologist
Cancer genomics	Developmental biology
Managing director	Post-doc
5	1
4	2
3	2
2	1
3	3
2	1
4	3
4	3
2	4
2	5
4	5
4	5
2	3
3	2
2	3
4	2
R, Perl, Bash	None
R	
Generate static	
I use it	I don't use it
I don't use it	I don't know what it
I don't know what it	I don't know what it
I don't know what it	I don't know what it
2	3
R10	**R11**

SUS scale recoded answers:

Table 21 Raw SUS scores and answers

Item 1	**Item 2**	**Item 3**	**Item 4**	**Item 5**	**Item 6**	**Item 7**	**Item 8**	**Item 9**	**Item 10**	**Total**	**Score**	**Learn**	**Use**
0	2	1	2	3	3	3	3	2	3	22	**55**	2.5	2.125
2	2	2	4	3	1	3	2	3	3	25	**62.5**	3.5	2.25
3	4	3	4	3	3	3	4	3	4	34	**85**	4	3.25
3	2	3	3	3	2	2	2	3	3	26	**65**	3	2.5
3	3	3	3	2	3	3	3	2	2	27	**67.5**	2.5	2.75

1	3	2
3	3	3
3	3	3
4	2	3
3	3	3
2	3	3
3	3	2
3	3	3
1	3	2
2	3	2
25	29	26
62.5	**72.5**	**65**
3	2.5	2.5
2.375	3	2.625

D. Improvements second cycle

Table 22 Improvements of second cycle

Comment	Update
More EDA	A new EDA panel was added and enables comparison of two data sets. Visualization with Boxplots, scatterplots of raw data or PCA are available.
Classification of methods	The first version of the prototype had a classification system of method based on the underlying normalization methods (retrieved from R vignettes). This was updated to a classification with the goal of each normalization package. In the prototype two goals present are normalization for outliers and PCA (RLOG transformation)
Data set check	Renders an HTML view of a data set to get a quick look on the content of a table. The first 5 rows and samples appear in a table.
Export/import CSV	The dynamic tables were adapted to enable export and import of csv files

E. Paper Proposition

Towards reusability of computational experiments

Capturing and sharing Research Objects from knowledge discovery processes

ABSTRACT

Calls for more reproducible research by sharing code and data are emitted in a large number of fields from biomedical science to signal processing. At the same time, the urge to solve data analysis bottlenecks in the biomedical field generates the need for more interactive data analytics solutions. These interactive solutions are oriented towards users on the wet lab side whereas bioinformatics favor custom analysis tools. In this position paper we elaborate on why Reproducible Research, by advocating more code and data sharing, misses the main issues in modern data analytics. We suggest new ways to design interactive tools embedding constraints of reusability with data exploration. Finally, we seek to integrate our solution with Research Objects as they are expected to bring promising advances in reusability and partial reproducibility of computational work.

Towards reusability of computational experiments
Capturing and sharing Research Objects from knowledge discovery processes

Armel Lefebvre[1], Wienand Omta[1] and Marco Spruit[1]
[1] Department of Information and Computer Sciences,, Utrecht University, Princetonplein 5, Utrecht, the Netherlands
armelefebvre@gmail.com, {w.a.omta, m.r.spruit}@uu.nl

Keywords: Knowledge Discovery, Reproducible Research, Bioinformatics, Research Objects, Software development.

Abstract: Calls for more reproducible research by sharing code and data are emitted in a large number of fields from biomedical science to signal processing. At the same time, the urge to solve data analysis bottlenecks in the biomedical field generates the need for more interactive data analytics solutions. These interactive solutions are oriented towards users on the wet lab side whereas bioinformatics favor custom analysis tools. In this position paper we elaborate on why Reproducible Research, by advocating more code and data sharing, misses the main issues in modern data analytics. We suggest new ways to design interactive tools embedding constraints of reusability with data exploration. Finally, we seek to integrate our solution with Research Objects as they are expected to bring promising advances in reusability and partial reproducibility of computational work.

1 INTRODUCTION

Over the last few years, calls from researchers defending better data and code sharing for computational experiments (CE) are propagated in high-ranked journals (McNutt, 2014; Peng, 2011).Usually grouped under Reproducible Research (RR), these invitations elevate reproducibility or replicability as a central key of the scientific method. One of the interpretation presents *reproduction* as an application, by independent researchers, of identical methods on identical data to obtain similar results whereas *replication* is similar except that different data is selected. According to RR proponents, benefits would be numerous.

First for verifying results of a published study (Peng, 2011). Second for reusing previous work and build new knowledge. While the latter brings a constructive and enriching dimension to reproducible science, the first one is clearly oriented to alleviating scientific misconduct, particularly in Life Sciences (Laine, Goodman, Griswold, & Sox, 2007).

Despite the fact that RR proponents are focused on suggesting to exchange code and data as a minimal threshold for "good science", they do not examine the methods used or people participating in CEs. Methods are not of interest to RR as the main focus lays on getting similar results for verification. In other words, results might be way off as long as they are verifiable. Hence, the end product of a CE is seen as a script, or package that should be made available by the authors of a paper as supplementary material.

The issue investigated in this work emerged from three phenomena: (1) the notorious increase of data generation and resource intensive analytics. Here in the biomedical domain, (2) ignorance about data generation processes and their impact in terms of modelization. For instance, the sequencing instruments and custom bioinformatics pipelines producing analytical data and how well they represent underlying biological facts and (3) non-specialists, not trained in data analytics, eager to participate in computationally intensive experiments but preferably via convenient end-user interfaces instead of custom scripts or programs.

The phenomena described above were observed during a design science research (DSR) (Hevner & Chatterjee, 2010) we conducted in the domain of biomedical genetics. Our research was focused on designing an interactive data mining tool for biologists to identify interesting outliers in RNA-Seq count tables. Additionally, information about methods or packages applied on data sets are made easily accessible via a web application. Ultimately, the goal is to seek how to facilitate access and reuse for *bioinformaticians* and *biologists* at the same time. After one design cycle of a technical artifact and its evaluation by three focus groups gathering biologists and bioinformaticians (n=15) we collected evidence against some practices proposed by Reproducible Research and suggest potentially fruitful improvements.

Indeed, r*eproducibility* of CEs should not be reduced to code and data sharing as it does not cover fundamental characteristics of modern data analysis in biology. We state that *web resources and their support for multiple representations that satisfy the interest of both types of users involved will have a positive impact on reproducibility by facilitating reusability first*.

As we will discuss in the next sections, reusability has its limits and challenges.

2 BACKGROUND

Two aspects of knowledge creation and sharing are presented. Together, they clarify what issues emerge from code and data sharing when all stakeholders involved in a CE are not considered. We make use of a standard knowledge cycle, the Integrated Knowledge Cycle (IKC) (Dalkir, 2005) to emphasize the issues of codification implied by Reproducible Research. In knowledge management, codification aims at making implicit knowledge (from an individual) available as an object that is separated from the individual (Hislop, 2005). In this context, objects are documents or an entry in a knowledge base which are available to people in an organization and are easily retrievable.

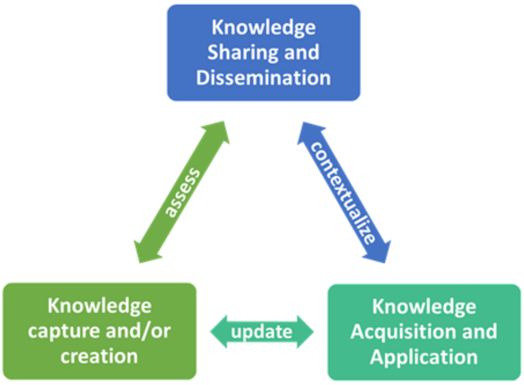

Knowledge Sharing and Dissemination
assess
contextualize
Knowledge capture and/or creation
update
Knowledge Acquisition and Application

Figure 69: Integrated Knowledge Cycle with three stages. The RRO-KDD covers Knowledge capture and Knowledge sharing (Dalkir, 2005).

The IKC is illustrated in Figure 69. We focus our discussion on the *knowledge capture and creation* and *knowledge sharing and dissemination* phases. The last phase *acquisition* is not discussed here as we believe it to be the role of academia or industry in general and not a particular process.

Codification may involve a selection of an information model (IM) to structure and report material about experiments such as the related projects or meta-data about patients' samples. The role and goals of Research Objects (RO) are introduced in section 2.1. Nevertheless, we already highlight that ROs check the availability of *sufficient information* using information models. The codification task has to comprehend what elements are part of an IM for a given experiment type.

We believe that this minimum information strategy is a positive choice for designing ROs as it fits the main outcomes of the focus groups we conducted (see section 4.1).

2.1 Reproducible Research

We start with Human-computer Interaction (HCI) which is the "study of the way in which computer technology influences human work and activities." (Dix, 2009). Knowledge discovery from databases (KDD) is defined by Fayyad as "the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" (Fayyad, Piatetsky-Shapiro, & Smyth, 1996).

The first aspect is that an end-user should be able to analyze data by using steps from the knowledge discovery process but interactively. This combination between KDD and human-computer interaction was theorized by Holzinger (2013). Tailored to the biomedical field, it argues that an end-user needs powerful visualization tools as much as data management and analytics capabilities. Holzinger also stresses the fact that reproducibility should be investigated further as it represents a major problem with data intensive experiments (Holzinger, 2013).

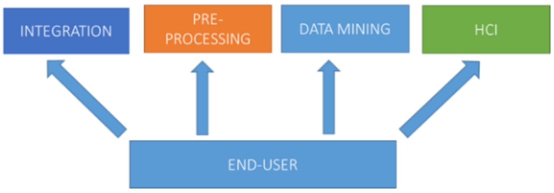


Figure 70: HCI-KDD based on Holzinger (2013).

As can be seen from Figure 70, the steps of the HCI-KDD are *integration*, *pre-processing* and *data mining*. *Integration* is the activity of merging structured or unstructured data sets. *Pre-processing* applies normalization or transformation techniques to make the data sets suitable for data analysis. *Data mining* is the design and application of algorithms to identify patterns, associations or outliers.

This process is combined with more user-friendly, powerful visualization and interaction tools and techniques investigated by the Human-Computer interaction field (HCI).

2.2 Reproducible Research

The second aspect is the need for more reproducibility of experiments which are conducted with computers. Here we integrate notions belonging to two approaches to reuse context and computational material.

On the one hand, based on *literate programming* (Knuth, 1984), dynamic documents (Pérez & Granger, 2007) and compendiums (Gentleman & Lang, 2007) constraint design choice to add *human* and *machine* readable context to executable code. Compendiums aggregate dynamic documents. Dynamic documents are executable files that contain code with descriptive information. They are currently available with authoring packages in R (Knitr, Sweave) or Python (IPython notebooks, Jupyter). Compendiums did not reach an implementation phase but include *transformations* of their content into *views* (Gentleman & Lang, 2007). It means that different stakeholders are taken into account in this preliminary design proposition. Access to a compendium is targeted to people with their own interests. For instance, students, researchers or teachers which may want to consult adapted content. Last, compendiums are seen as *executable* entities able to redo an analysis and support meta-information.

On the other hand, an ontology based approach for dissemination of reusable components is assured by semantically enriched objects aggregating resources about the context of an experiment and its material. These are called Research Objects (RO) (Bechhofer et al., 2013). Tentative RO designs are extending the Open Archives Initiative Object Reuse and Exchange (OAI-ORE) ontology (Belhajjame et al., 2014).

Initial attempts are centered on workflows. They aim at making these workflows exchangeable, reusable and reproducible (Belhajjame et al., 2014). To achieve this, additional information, about hypotheses, for instance, are annotated and "packaged" before being uploaded to a RO repository (see http://www.researchobject.org/).

3 DISCUSSION

As we noticed, the fact that one end-user deals with each step is, at least, a very optimistic view on data analytics. The HCI-KDD process implemented in the prototype was discussed among participants. The questions were oriented to the flow of analysis and presence or absence of components (e.g. charts, packages, result tables, context…) in the interface. Additionally, a survey was answered by 11 respondents (n=11) about how they are dealing with data and Reproducible Research. The results of this survey are discussed through the following subsections.

3.1 Focus groups result

Inside our three focus groups we divide participants according to their main interests, i.e. bioinformaticians and biologists.

For the first type of participants, bioinformaticians, a friendly user interface is rejected. Scripts are preferred for analyzing data. Regarding methods applied, a participant indicated that a method is sometimes selected because “it works” and is not a matter of “hidden” assumptions. By assumption we refer to *prior knowledge of the state of the world* embedded in packages or statistical models. Not being aware of them makes a package acting as a “black-box” with unknown consequences on the rest of the processing. Some participants suggested to only provide custom-packages instead of original packages (in R or Python) to biologists. These packages would be prepared by bioinformaticians and hide all the complexity of models or data preprocessing.

For the second type of participants, biologists, they estimated the presence of such methods as appropriate. The indications given on the website (package name, version, reference paper, running environment and online documentation) are sufficient if kept up-to-date. The web interface offered the possibility to apply different methods on the same data set. This was judged as beneficial because the influence of a choice could be assessed by the user interactively. In that case, another concern raised by bioinformaticians is about the *interpretation* of results by users that would not be trained in statistics.

Table 23: Conclusion of Focus Groups. The maximum support is 3, the number of organized focus groups.

Feature	**Support**
Method recommendation	3
More Exploratory Data Analysis	3
Minimal Context integration	3
Workflows	2
Search in data sets	2
Custom package	2
Comparison, merge, quality checks	2
Dynamic documents	2

Regarding reproducibility, the lab part of an experiment has strong influences on the rest of the pipeline and it is perceived as challenging to integrate in the tool. Efforts for improving reproducibility are welcome but full reproducibility is impossible, as indicated by participants in the third focus group.

In summary we can conclude that biologists are indeed more on the HCI part and bioinformatics on KDD but these two aspects are still to be integrated properly as our interest is also to reuse outputs of all these steps. A global overview of features and their support is given in Table 23. Support indicates how many focus groups agreed that these features are required in a data mining tool. Discussions were backed by screenshots or short demonstrations of the running tool.

3.2 Code and data for verification

It is the view of Peng (2011) that executable code and data form a gold standard of reproducible research. We argue that these elements are not of interest for each important type of stakeholder involved in a computational experiment. We may admit though that what the author tries to achieve is a minimal level of *reproducibility* for *verification* purposes. The idea is that a reviewer would carefully inspect code shared with a paper, e.g. as an R package on Bioconductor. With that package, the entire computational workflow is *runnable* and shows figures that are identical to their online or printed counterpart.

But as even noticed by Peng (2011), papers validating previous work are rarely acclaimed by publishers which expect “new” knowledge to be submitted. This may be an explanation while results from our survey showed a poor interest in full replication. On a scale from 1 (never) to 5 (always). The need for full replication has a Mode of 2 (Median=2). Partial replication did slightly better with a Mode of 3 (Median=3).

Additionally, features discussed among practitioners earlier and listed in Table 23 are not fundamentally concerned with reproducibility issues. Still, we may imply that “method recommendation” and “custom package” adhere more to *reusability* principles.

3.3 Reusability and interactivity

Regarding Research Objects, they sometimes appear to be developed as external solutions. Indeed, we find traces of RO management tools in the wf4ever repository of GitHub. We would lose a major group of researchers if the goal of an application is to purely manage research objects. Instead, the software application should produce resources that might be automatically aggregated in a RO. This is a transparent manner for users more interested in advanced visualization capabilities than how it works at the package or environment level.

Therefore, we claim that Research Objects could be a hidden component of any interactive mining tool. By doing this, we encourage RO generation and usage without transforming such tools in a “reproducibility manager” for users interested in getting precious insights from their experiments. Exaggerating any requirement of RO management for these stakeholders will most probably result in a rejection of the entire application. This could be achieved by automatically extracting information from earlier processing stages and intermediate data sets in the analysis flow.

3.4 Resources and Representations

An interesting proposal in compendium design was the notion of *transformer*. We present it in this work as the creation of a *representation* (or view) from a single *resource*. A *resource* is an object of interest whereas a representation is a usable form of a resource which corresponds to the consumer’s interest. We designate by consumers both human and machine readers or interpreters.

In the RO world, it implies to work on ontologies and machine readable standards. For biologists, it means that a *chart* resource has to render a dynamic representation. We can imagine that after exchanging a RO, we find a *data object* resource and a *chart* resource. A chart shows the content of a data object as, for instance, a scatterplot. We expect an end-user to be willing to select parts of this scatterplot, zoom-in or display labels. We also expect that this *chart resource* is identical to what was generated by a team of researchers which created this RO and included a chart resource inside.

As we show in the next section, open source technologies for visualization "as a resource" exist and are under heavy development. They are able to create Json or html/JS serialization of a chart resource while providing enough interactivity for end-users.

4 SOLUTION

The evaluation of our prototype yielded limitations of both HCI-KDD and current practices defended by Reproducible Research. Hence, we suggest an improved knowledge discovery process embedding the HCI-KDD in an extended process named RRO-KDD. RRO-KDD merges highly interactive KDD with reusability of web resources.

To maximize the validity of our design proposition, a design science research method was applied (Hevner & Chatterjee, 2010). To begin with, we present the three elements forming building pieces of a DSR framework. First, *environment* lists what the stakeholders, domain of research and problems are. This guarantees that the DSR is actually relevant and benefits to the stakeholders by improving a situation with information technologies.

Next, *design science research* elaborates on the artifacts that are designed. For this study two artifacts were designed. 1) A socio-technical artifact which is a prototype application for interactive data mining. 2) A process artifact which is based on the feedback collected about the prototype. The process artifact is a design proposition. It merges the HCI-KDD process with research objects. We called this process artifact a RRO-KDD for Reproducible Resource Oriented Knowledge Discovery from Data.

After each design cycle, the prototype is evaluated. Because this paper reports only on the first cycle, there was only one evaluation with focus groups. Inside each design cycle we have sub-activities. The *problem specification* resulted from a literature review and meetings with experts in biomedical genetics. The other steps found in design science research are *Intervention*, *Evaluation* and *Reflection.*

Each of them are described in the next subsections.

4.1 Specification

The problem addressed in this work encompasses reproducibility and visualization for researchers in biology who are collaborating with bioinformaticians. As explained in the background section, computational experiments are not only conducted on the bioinformatics side of data analysis. Hence, an application enabling *self-service* data analytics has additional constraints. *Self-service* is understood as letting users perform analytics tasks without advanced knowledge of programming or statistical modelization.

4.1 Intervention

As technical outcome of the DSR we conducted, a prototype was developed and deployed in a research lab for structural genomics, UMC Utrecht, the Netherlands.

The prototype is focused on RNA-Seq data, a technique used to measure gene expression (Wang, Gerstein, & Snyder, 2009). For instance, this technique is applied to measure whether there is a significant difference between expression levels in tumor and healthy tissues of cancer patients. Rising expression levels might indicate an extra "activity" of one or more genes that may be responsible for driving tumor growth.

The prototype started from the HCI-KDD process by implementing interactive visualization capabilities together with methods to pre-process and mine data sets. Pre-processing consisted in *normalization* and *transformation* of table of counts generated by RNA-Seq technologies and tools. A table of counts has samples of patients in columns and a list of genes as rows (60 000 in the files used).

Table 24: An example of table of expression counts.

Genes/patient tissue	**P1.Tumor**	**P1. Healthy**	**P2. Tumor**
Gene 1	120	144	90
Gene 2	30	20	1200
Gene$_i$	…	…	…

Table 24 shows a fictional data set that is handled by the prototype. This table is the result of a *bioinformatics pipeline*. Hence, analytical data is generated by various levels of data processing from raw DNA sequence quality checks to counting how many RNA fragments found in a patient tissue overlap a gene. This information is not exploitable via the user interface as it is not contained in the uploaded files yet. To achieve this, other data files are necessary which justify the presence of an *integration* step.

Via the web interface, users start with these processed data sets in a *virtual experiment* (gathering data and contextual information). Then a possibility is offered to *normalize* or *transform* data sets by calling packages from Bioconductor. Normalization is an important pre-processing task to make samples comparable due to the presence of (technical) biases in the raw data.

4.2 Evaluation

Exploratory focus groups with biologists and bioinformaticians provided input for conducting additional iterations, similar to an agile approach. From requirements and discussions with specialists a set of functionalities for KDD and visualization were implemented. The facet of RR was imposed as it was not a primary requirement from the field experts. Hence, design choices for RR were inspired by previously described literature about compendiums and ROs.

Next, three confirmatory focus groups (Tremblay, Hevner, & Berndt, 2010) invited bioinformaticians and biologists to discuss about the prototype and judge the applicability of the KDD steps implemented. We addressed results obtained from the focus groups in section 3. These results are further processed is section 4.4. We present a design proposition which is an outcome of the evaluation of the prototype. Furthermore, our design proposition covers architectural choices which are mainly grounded in the web architecture.

4.3 Reflection

The lessons learned from our DSR are described in the RRO-KDD process. We processed the input of three confirmatory focus groups with 15 participants. We described the results earlier and elaborate on their processing further in the next section.

4.4 RRO-KDD Process

In Figure 3, the process is modeled with its related "deliverables" in a so-called process-deliverable diagram (PDD) (Weerd & Brinkkemper, 2008). Here, the elements of the HCI-KDD process are integrated with contextual and technological outputs. These outputs are directed to reusability of previous experiment code, data and methods. Below, we shortly describe the steps and deliverables:

1) Understand is an activity where sufficient description of the data sets are provided. For instance, information about instruments, sequencing platforms, sample preparation. It builds a *container* for an experiment which is denoted by *virtual experiment*. Virtual experiments are uniquely

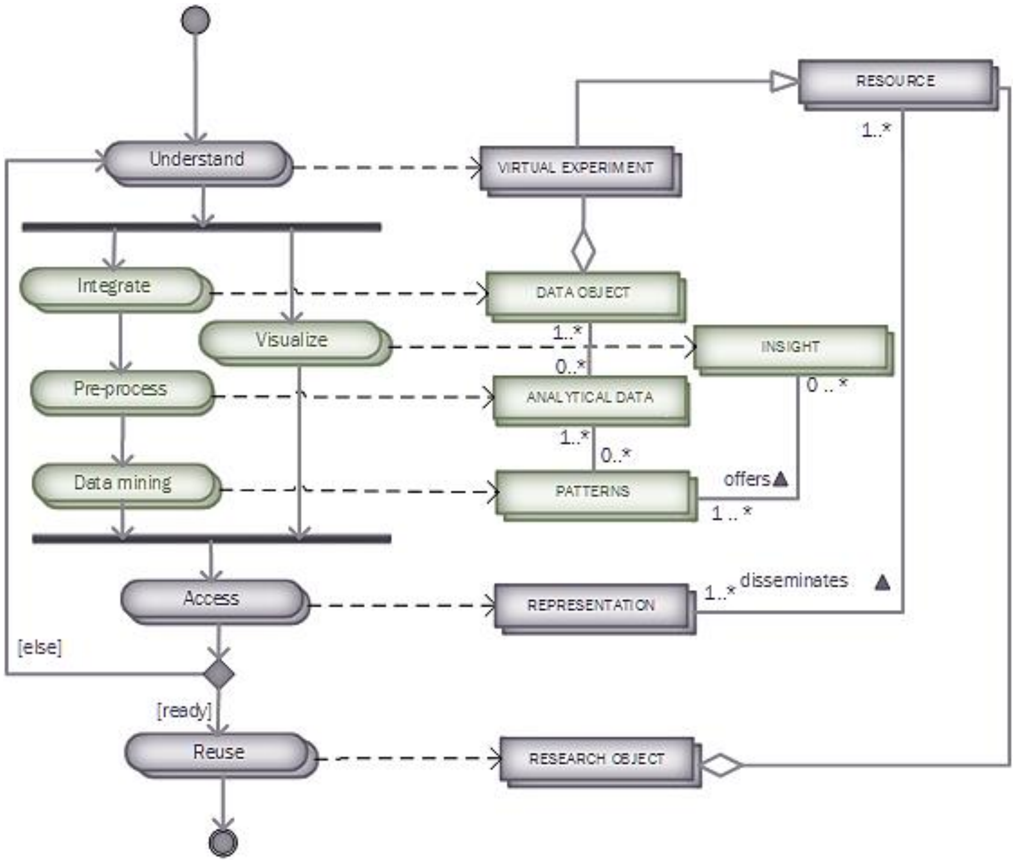


Figure 71: RRO-KDD proposition.

identified aggregation of resources and group data sets together with context and methods.

2) Integrate, pre-process and data mining are the steps elaborated by the HCI-KDD process. Visualization is an activity that occurs in parallel to KDD and enables to get insight of what happens at each step. For instance, it helps the users to judge the impact of pre-processing methods on the data set. Activity *Integrate* gives data objects. These are entities uniquely identifying data sets in accordance to FAIR data management principles. *Pre-process* will normalize or transform these integrated data sets in analytical data which is more easily interpretable than raw data, e.g. from sequencing instruments. Finally, data mining results find useful patterns from data, according to Fayyad's definition (Fayyad et al., 1996). Visualization is here a subpart of the whole HCI field of research as it was not extensively investigated in this work.

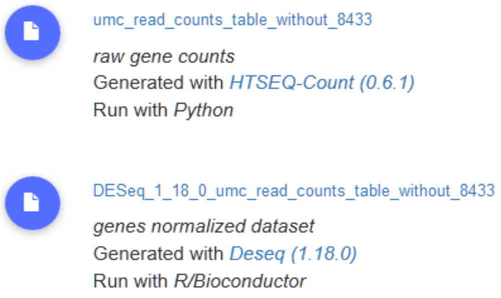
3) Visualization has a deliverable called insight, which informs researchers on patterns, scores or relations in their data on an interactive manner. Interactive plots were rendered with *bokeh*, a python library for creating browser compatible visualizations. The prototype is limited to scatter plots, boxplots and heat maps.

4) Access presents previous, interactively created components of an experiment (like charts and new data objects) as REST resources that might be accessed without the user interface via REST clients. This activity combines access from the user interface and from a web API to satisfy both type of users.

5) These resources, aggregated in a virtual experiment can be semantically enriched for reuse as ROs. This is made possible because each component is uniquely identified and accessible via a programming interface. As an example, a *mining task* created by a biologist is reusable via a RO with its unique identifier. A *GET* request will trigger the execution of a method on a data object with all the recorded parameters of the given mining task.

We illustrate this principle with an outlier mining task M on a data set D, identified as a data object, where a preprocessing method P was applied. P has some prerecorded parameters in a database. The identifier of this task is *M_P_D_GUID*. A user can select this task in a list or get the results via the Web API by using the same unique ID. In a RO, one will find among other components the executed task with its identifier.

A virtual experiment is a resource. It can be accessed and render different representations. With the *content negotiation* mechanism implemented in HTTP. One can imagine to retrieve a dynamic plot or the data behind the plot by setting the accept header appropriately. In that case, for instance, the data is retrieved with accept set on application/Json and the dynamic plot with text/html. It implies that the same resource is used in both cases and avoid mistakes such as selecting different data objects to retrieve raw data or to visualize it. Figure 72 illustrates how these identifiers are selected in a list of data objects in our prototype.



umc_read_counts_table_without_8433

raw gene counts

Generated with *HTSEQ-Count (0.6.1)*

Run with *Python*

DESeq_1_18_0_umc_read_counts_table_without_8433

genes normalized dataset

Generated with *Deseq (1.18.0)*

Run with *R/Bioconductor*

Figure 72: Selection of data objects with identifiers containing methods applied as prefixes.

The code of the prototype is hosted on GitHub under MIT license and is available here: https://github.com/armell/RNASEqTool.

5 CONCLUSION

Results suggest that reproducibility cannot be reduced to data and code sharing and that the field of biomedical genetics suffers from a lack of software solutions that are both satisfactory for bioinformaticians and biologists who are mutually engaged in CEs. There are overlapping data analytics practices but also serious apprehensions from bioinformaticians to offer such a type of application to biologists if they exceed data visualization.

Despite these concerns, we found that there is gap to fill both in terms of data analytics and reuse of previous work. The suggested approach complies with the Minimum information models handled in Microarray analysis or more recently in Research Objects design. Capture of footprints (*resources)* during the interactive data analysis phase enables reusable practices by *content negotiation* and REST principles.

As we have seen biologists were more inclined to ask more visualization capabilities whereas bioinformaticians expect a solution where scripting or custom data processing is allowed. Unique identifier of resources and platform-independent information exchange via REST enables this. Nevertheless, HCI alone for biologists is not satisfactory as they want to query data and compare the impact of different methods. These comparisons require pre-processing and mining.

Reusability of data, workflows or parts of experiments seems to be more interesting for the two types of end-users which evaluated the artifact than reproducibility. For this study, the first cycle of a design science research in biomedical genetics for data mining and reproducible research was evaluated by experts in three confirmatory focus groups.

All participants worked in a Dutch academic environment which may bias the answers towards one single country. Nevertheless, this limitation is slightly mitigated as we assume that the way of working in genomics and the techniques applied are similar across labs and research units.

6 FUTURE WORK

The suggested RRO-KDD is still in a design proposition phase that needs to be evaluated in other settings and the interest in sharing Research Objects must be assessed. For this assessment, the mining tools have to be upgraded and provide more realistic possibilities to exchange and reuse virtual experiments and their components.

In addition, extending the RRO-KDD to distributed systems will have similar problems encountered in previous studies and known as *workflow decay*. This issue still holds in the RRO-KDD context which is built around web services and URLs that may be inactive after some time. Permanent Identifiers may moderate accessibility issues but not the support of data objects or remote implementations of analysis packages.

Recommendations to face these issues are an integration with virtual environments or containers (e.g. Docker), dynamic documents and proper data management solutions. More research on integrating virtual containers for reusability of computational experiments for bioinformaticians and biologists is needed. A careful assessment on how dynamic documents could reflect what a biologist performed interactively and how to communicate it to bioinformaticians might be valuable. These documents could also play a role for bioinformaticians to understand what decisions were taken by biologists processing data via a user-friendly interface. Data management is an aspect out of the scope of this paper but is mandatory for the proper functioning of reusable outputs from an interactive data analysis tool.

These investigations should be made by effectively combining HCI and KDD as suggested by Holzinger. But the multiplicity of actors, analysis tools and techniques remains a great challenge first for reusability then for reproducibility.

Hence, reproducibility arguments in literature should be replaced by better designs for reusability in IT solutions, at least for enhancing collaboration between bioinformatics and biology. *Reusability* is broader than reproducibility as it enables *repurposing* of previous work and, in essence, *reproducibility*.

ACKNOWLEDGEMENTS

Our thanks go to Dr. Wigard Kloosterman (UMCU) and his team for hosting us, providing any resource to conduct our research and assisting at the demo sessions.

REFERENCES

Bechhofer, S., Buchan, I., De Roure, D., Missier, P., Ainsworth, J., Bhagat, J., … Goble, C. (2013). Why linked data is not enough for scientists. *Future Generation Computer Systems*, *29*(2), 599–611. doi:10.1016/j.future.2011.08.004

Belhajjame, K., Zhao, J., Garijo, D., Hettne, K., Palma, R., Corcho, Ó., … Goble, C. (2014). The Research Object Suite of Ontologies: Sharing and Exchanging Research Data and Methods on the Open Web. *arXiv Preprint arXiv: 1401.4307*. Retrieved from http://arxiv.org/abs/1401.4307

Dalkir, K. (2005). *Knowledge Management in Theory and Practice*. *Knowledge Management* (Vol. 4).

Dix, A. (2009). Human-Computer Interaction. In L. LIU & M. T. ÖZSU (Eds.), *Encyclopedia of Database Systems SE - 192* (pp. 1327–1331). Springer US. doi:10.1007/978-0-387-39940-9_192

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). Knowledge Discovery and Data Minin: Towards a Unifying Framework. In *Proc 2nd Int Conf on Knowledge Discovery and Data Mining Portland OR* (pp. 82–88).

Gentleman, R., & Lang, D. (2007). Statistical analyses and reproducible research. *Journal of Computational and …*, *16*(1), 1–23. Retrieved from http://amstat.tandfonline.com/doi/abs/10.1198/106186007X178663

Hevner, A., & Chatterjee, S. (2010). *Design research in information systems*. Springer New York.

Hislop, D. (2005). *Knowledge management in organizations: A critical introduction*. *Management Learning* (Vol. 36).

Holzinger, A. (2013). Human-Computer Interaction and Knowledge Discovery (HCI-KDD): What is the benefit of bringing those two fields to work together? In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 8127 LNCS, pp. 319–328). doi:10.1007/978-3-642-40511-2_22

Knuth, D. E. (1984). Literate Programming. *The Computer Journal*, *27*(2), 97–111. doi:10.1093/comjnl/27.2.97

Laine, C., Goodman, S. N., Griswold, M. E., & Sox, H. C. (2007). Reproducible Research : Moving toward Research the Public Can Really Trust. *Annals of Internal Medicine*, *146*(6), 450–453. Retrieved from http://annals.org/article.aspx?articleid=733696

McNutt, M. (2014). Journals unite for reproducibility. *Science*, *346*(6210), 679–679.

Peng, R. D. (2011). Reproducible research in computational science. *Science (New York, N.Y.)*, *334*(6060), 1226–7. doi:10.1126/science.1213847

Pérez, F., & Granger, B. E. (2007). IPython: A system for interactive scientific computing. *Computing in Science and Engineering*, *9*, 21–29. doi:10.1109/MCSE.2007.53

Tremblay, M. C., Hevner, A. R., & Berndt, D. J. (2010). The Use of Focus Groups in Design Science Research. In *Design Research in Information Systems* (pp. 121–143). doi:10.1007/978-1-4419-5653-8_10

Weerd, I. Van De, & Brinkkemper, S. (2008). Meta-modeling for situational analysis and design methods. *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications*, 38–58.