Short Communication

Development of Applications for Interactive and Reproducible Research: a Case Study

Federico Marini\(^1\), Harald Binder\(^1\)

\(^1\)Institute of Medical Biostatistics, Epidemiology and Informatics (IMBEI), University Medical Center, Mainz, 55131, Germany
\(^2\)Center for Thrombosis and Hemostasis (CTH), University Medical Center, Mainz, 55131, Germany

Correspondence: marinif@uni-mainz.de

Received 2016-09-30; Accepted 2016-10-25

ABSTRACT

For a proper understanding of the organization and regulation of gene expression, the computational analysis is an essential component of the scientific workflow, and this is particularly true in the fields of biostatistics and bioinformatics. Interactivity and reproducibility are two highly relevant features to consider when adopting or designing a tool, and often they can not be provided simultaneously.

In this work, we address the issue of developing a framework that can provide interactive analysis, in order to allow experimentalists to fully exploit advanced software tools, as well as reproducibility as an internal validation of the analysis steps, by providing the underlying code and data in such a way that enables the re-creation of the results, and also constitutes a didactic tool for the life scientist.

We illustrate this paradigm with the help of the R/Bioconductor package pcaExplorer, designed as a practical companion for interactive and reproducible exploratory data analysis for high dimensional data (e.g. RNA-seq), and highlight some of the features that are provided in the software.

KEYWORDS

Interactivity; Reproducibility of Results; Visualization; Exploratory Data Analysis; RNA-seq

INTRODUCTION

Computational analysis has become a standard component in the workflow of each scientific effort that aims to dissect the complex organization and regulation of gene expression.

When addressing the challenges deriving from the specific scientific question, the adoption or development of dedicated tools is an essential choice for the life scientist, and this holds true also for the experienced data analyst. Two fundamental aspects that characterize software tools are interactivity and reproducibility, and unfortunately most of the existing software cannot provide these two features simultaneously.

A framework which leverages on interactive analysis fully empowers domain experts in extracting information from the complex data they generated, e.g. by efficiently exploiting visualization tools as a key for knowledge generation and sharing of scientific findings [1]. Life scientists aim to analyze and visualize their datasets with interactive end-user interfaces, as they are easier to adopt than a fully scripted paradigm. These datasets are often the output of one or more bioinformatics pipelines with peculiarities and complications that may be unknown or too complex to these end-users. On the other hand, journal editors are legitimately asking for increasingly more detailed information about the methods or data processed during a study [2–4]. Thus, it is essential that biologists and data analysts reach a common ground to overcome the hurdles posed by advanced technologies, such as next-generation sequencing machines.

Reproducibility, which usually comes along with transparency and allows for independent verification, is becoming in the recent years a common goal for the scientific community, as such an approach ensures the degree of objective examination which is vital to achieve the quality control that justifies the public trust in science [5]. In a commonly accepted definition, it is defined as re-performing the same analysis with the same code, where the analyst can be different, but population, data and analysis plan do not change [6].

A minimal standard for reproducibility is guaranteed by the assembly of code and data in such a way that any other scientist could re-create all of the results (e.g., tables and figures of a publication) - a condition that yet do not automatically imply the correctness of a work (e.g. buggy code, poorly performing methods). Whereas the end user is typically not expected to be proficient at coding, there are doubtless benefits in implementing a workflow to make results reproducible [7], as this will make the analytic life easier: problems or questions, that might occur or arise anywhere in the downstream part of the process, will be more easily identified, corrected or explained. Moreover, aiming towards reproducible research can be also seen as a driving force to design better analytic systems, which should organize and automate most aspects of the workflow, and deliver for free the reusability of the own analytic work [8]. This approach has been highlighted in recent editorials, by pointing at the advantages of review, replication, reuse and recognition as incentives to provide the code for generating scientific results [9].

This work will address the issue of identifying and developing a framework that is able to provide interactive and reproducible analysis, which would grant the advantages of each aspect, and at the same time...
guarantee a level of user-friendliness that can reach different levels of expertise, as well as serve as a didactic tool for learning the recipes to turn raw data into publication-ready output.

We developed several R/Bioconductor packages that fulfill these requirements, among others flowcatchR [10] and pcaExplorer [11]. flowcatchR is an integrated workflow solution (R package [12], interactive Shiny application [13], IPython/Jupyter notebooks [14]) to analyze imaging data, focused on tracking flowing blood cells, whereas pcaExplorer provides the functionality for interactive exploration of RNA-seq data [15] based on principal components analysis [16].

In this work, we will use the latter to discuss the possibilities and the necessary components involved in the development of this software design model.

**CASE STUDY**

**Implementation**

Due to its flexibility and being well established in the field of computational biology, the R programming language [12] was chosen for the development of the software package. Furthermore, the R environment offers the adoption of the Shiny framework [13], which allows for a lightweight and effective creation of interactive graphical user interfaces, and exploits reactive programming [17] to make computation of the underlying R objects efficient.

The pcaExplorer package itself is included in the Bioconductor project [18] (Release 3.3), and builds upon a variety of other available packages. As a main function for the package, the execution of `pcaExplorer()` will suffice to launch the application in a web browser.

The reproducibility aspect is dealt with by leveraging on the R packages knitr and rmarkdown [19, 20], that together constitute the foundation of producing dynamic documents with R. The paradigm of literate programming allows the generation of reports where code, output and plain text are combined in a single output, with markdown as a lightweight and simple language for markup and formatting. This functionality is provided in a dedicated tab of the application, where a template report is provided as a default document, and this can be edited according to the needs of the user. The generated report can be previewed and downloaded, and then readily shared.

While running, the current settings, parameters, widget status as well as the underlying data are stored as reactive values by the application. These are by definition values whose results can change during runtime, are accessed for the report generation, and can also be exported via the state saving functionality, either to the global environment or as binary data, for further reuse or sharing with collaborators.

**Tool Description**

In applications such as RNA-seq, the tens of millions of reads generated to probe the expression levels of the transcriptome features of interest [15] are subsequently transformed into high-dimensional matrices, using tools such as featureCounts [21] or HTSeq [22], where each element represents the expression value for a gene in a specific sample.

A vast number of tools and software packages have already been developed for the many workflows where such datasets are involved (e.g. identification of differentially expressed genes [23, 24], study of alternative splicing events, novel transcript discovery, gene fusion, RNA editing, see [25] for a comprehensive overview).

The pcaExplorer package is intended to ideally accompany the analysis for any transcriptomic dataset, and its focus is set on the exploratory data analysis step, which is common to all the research questions listed above. Despite its relevance for the reliability of the generated results, it is often neglected; one reason could be due to the required proficiency in programming languages required by many of its aspects.

Among the many techniques adopted for exploring gene expression profiles, Principal Components Analysis (PCA) [16] is often used to obtain a dimension-reduced overview of the data. Even if some tools are available for performing such operations (e.g. ClustVis [26], BiplotGUI [27]), none of them feature an interactive analysis, fully integrated in Bioconductor, while also providing the basis for generating a reproducible analysis [5, 28].

The pcaExplorer app (screenshots shown in Fig.1) is structured as a dashboard, which allows for an efficient and intuitive exploration of the data, where each module is a tab of the main panel. The user can either upload the count and experimental data from text tabular files, or alternatively provide the R objects as parameters, as also detailed in the package vignette. Thanks to the Shiny framework, the user can brush (i.e. select a plot area) with the mouse to zoom in/select subsets of interest, click to generate additional visualizations and tables for further exploration. Live updates of the produced output are performed efficiently thanks to reactivity, and remarkably each plot/interactive table element can be readily exported in publication-ready quality via a simple mouse click.

The Report Editor tab (Fig.1, lower panel) provides the toolset for enabling reproducible research in the exploratory analysis, by capturing the current state of the ongoing session, combining it with a pre-defined analysis template. Furthermore, it allows to edit the code in the embedded editor via the shinyAce package [29], thus with optional support for code completion. The interactive HTML report is produced exploiting the knitr and rmarkdown packages, previewed in the app, and can be subsequently exported with simple mouse clicks. Its content is structured following the tabs of the applications, thus including an overview on the data, the focused views on samples and genes, as well as summary information on the functional annotation of principal components. Source code, output and narrated text are thus easily reproduced, and this step can be enhanced through the state saving functionality accessible from the app task menu.

The pcaExplorer package vignette describes in an extended way its functionality, alongside the additional functions which are provided by the software. Notably,
Figure 1: Screenshots of the pcaExplorer application. Upper panel: the Samples View tab, where the user brushes with the mouse to zoom in the PCA plot. Lower panel: the Report Editor tab, with markdown options on top, and an excerpt of the code for the report in the editor. On the left side of each panel, the sidebar menu offers a large set of options to control the generated output.
pcaExplorer can also be deployed as a web application on a Shiny Server, such that experimental users can explore their data without the need of software installation.

**CONCLUSION**

After identifying interactivity and reproducibility as two essential features for software that aims to effectively extract knowledge in the context of gene expression regulation, we proceeded in finding and further developing a framework that could guarantee the advantages of both aspects.

Taking the example of the pcaExplorer package we developed, this work presents an overview of exploratory data analysis for RNA-seq data, completely integrated in the R/Bioconductor ecosystem. Such an approach constitutes a practical and solid solution to the challenge of efficiently explore, visualize and disseminate the wealth of information contained in high dimensional datasets. Potentially, this tool can be directly adopted for the analysis of cutting edge experimental data, such as single cell RNA-seq.

We expect that such a model for tool development can be also adopted at different steps for transcriptome analysis (e.g. for inspecting the results of differential expression), or for other fields in computational biology (e.g. bioimaging), to provide a rigorous and user friendly environment for customizable analysis tailored to the project needs.

**ACKNOWLEDGEMENTS**

The work of FM fulfills part of the requirements of his PhD and is supported by the CTH, Mainz, Germany.

**AUTHOR CONTRIBUTIONS**

FM developed the package, implemented the algorithms and wrote the manuscript. HB supervised the project and wrote the manuscript.

**CONFLICT OF INTEREST**

The authors declare no conflict of interest.

**REFERENCES**


