Towards Reproducible Bioinformatics: The OpenBio-C Scientific Workflow Environment

Alexandros Kanterakis[1], Galateia Iatraki[1], Konstantina Pityanou[1,2], Lefteris Koumakis[1],Nikos Kanakaris[3], Nikos Karacapilidis[3], George Potamias[2]

[1]Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH), Heraklion, Greece
{kantale,giatraki,koumakis,potamias)@ics.forth.gr

[2]Dept. of Electrical and Computer Engineering, Hellenic Mediterranean University Heraklion, Crete, Greece
kwna.pitianou@gmail.com

[3]Industrial Management and Information Systems Lab, MEAD, University of Patras, Patras, Greece
nkanakaris@upnet.gr, karacap@upatras.gr

Abstract—**The highly competitive environment of post-genomics biomedical research is pushing towards the production of highly qualified results and their utilization in clinical practice. Due to this pressure, an important factor of scientific progress has been underestimated namely, the reproducibility of the results. To this end, it is critical to design and implement computational platforms that enable seamless and unhindered access to distributed bio-data, software and computational infrastructures. Furthermore, the need to support collaboration and form synergistic activities between the engaged researchers is also raised, as the mean to achieve consensus on respective scientific discoveries. The aforementioned needs and requirements underlie the motivations for the development of OpenBio-C workflow environment, and determine its fundamental objectives. The insufficiencies of current workflow editing and execution environments are explored, and the key-directions to advance over an environment that supports reproducibility are stated. The basic components and functionality of the OpenBio-C initial implementation (beta version) are detailed**.

Keywords-bioinformatics, scientific workflows, collaborative systems, open science

I. Introduction

With the global scientific output doubling every nine years [1], there is a question whether this increase has a clear impact on the "real" and actionable knowledge growth. The majority of publications are characterized as "background noise" and researchers are struggling to "separate the wheat from the chaff" [2]. Moreover, researchers still use generic Google-like searches to locate useful tools and analysis pipelines and rely on specialized web forums to seek technical advices. It is not an exaggeration to claim that science on that matter has not changed over the last 20 years. This is surprising, however, given that the results of most of the millions of publications in experimental sciences have been generated through an *analysis pipeline* of some kind. In fact, these pipelines are rarely publicly available. The temporal decay and eventual loss of these pipelines, in contrast to the almost ever-preserved and well-indexed scientific papers, constitutes a *major knowledge and cultural loss* not only for the scientific community but, for the society as a whole.

Reproducibility "now". It is reported that published and highly-ranked biomedical research results are very often not *reproducible* [3]. In a relevant investigation, about 92% (11/12) of interviewed researchers could not reproduce results even if the methods presented in the original papers were exactly replicated [4]. In addition, there are estimates that preclinical research is dominated by more than 50% of irreproducible results at a cost of about 28 billion dollars [5]. We argue that this unnatural separation between scientific reports and analysis pipelines is one of the origins of the *reproducibility crisis* [6]. Taking the above into account, one may even state that "*reproduce or perish*" should be the new motto for contemporary science [7].

Data, data everywhere! The daily production of post-genomic data worldwide outpaces even traditional "big data" domains, such as astronomy and Twitter, in terms of acquisition, storage, distribution, and analysis of respective datasets [8]. Even if advanced computing environments and solutions for analyzing large amounts of data are available (e.g., distributed computing applications, bulk data storage technologies, cloud environments), their application to post-genomics data is still limited. This is due to difficulties in accessing relevant computational infrastructures, to their adaptation and customization complexity, as well as to the needed computer skills. In addition, factors related to security as well as to ethical and legal issues, constrain biomedical researchers to adopt these technologies [9].

Interpretable analytics. However, the notion of "big data" is expanded to refer not only to the data volume itself, but to the ability to analyze and interpret those data. Especially in the field of post-genomics the rate of data generation outperforms the respective figures for their analysis and effective interpretation, making the production of interpretable results largely unfulfilled [10]. In domains where reliability and transparency are of critical importance, the case of *precision medicine* studies, the need is even more intense [11], [12]. We could say that in the current biomedical research landscape the participating researchers often do not know (*how*) to analyze and fail to make sense (*what*) of the scientific findings and discoveries. This makes it harder to *translate* scientific discoveries into credible clinical protocols and recommendations. Evidently, from about 150,000 scientific publications reporting on the discovery of new biomarkers, only 100 of them have found their way to the clinic! [13]. Without exaggeration, one would say that researchers of interdisciplinary fields, such as that of post-genomics biomedical research, experience a "*lost in translation*" situation! [14].

The above pose a major challenge to the contemporary bioinformatics community namely, to establish an *extrovert research ecosystem that enables open-science activities and ease the seamless exploitation of data and tools* (ec.europa.eu/research/openscience). This is exactly the motivation and the vision underlying the OpenBio-C bioinformatics platform presented in this paper.

II. State-Of-the-Art Bioinformatics Environments and Collaboration Support Tools

A. *Bioinformatics Workflow Environments*

Extensive reviews of bioinformatics *scientific workflow management systems* (SWfMS) already exist in the literature [15]–[18]. In this section, we briefly report on the most well-known and widely utilized ones.

Galaxy. It is probably the most successful SWfMS environment [19]. Galaxy is supported by a strong bioinformatics community with a fairly large number of distributed server installations in many research institutions worldwide. In spite of its success, Galaxy is still lacking in several capabilities such as the ability of users to collaborate during the composition and sharing of workflows, as well as the ability to evaluate and score the delivered workflows by exchanging ideas, suggestions and alternative approaches.

Taverna. It is the second most utilized SWfMS by the bioinformatics community [20]. Taverna has, at least in relation to Galaxy, limited use in post-genomics research. The reason is that it is based on an offline and relatively more complex workflow synthesis framework.

Programmatic packages. In addition to integrated *scientific workflow* (SWf) synthesis and execution environments we should also refer to solutions that are based on programming languages and provide the ability to synthesize workflows through relevant programmatic scripts. Such solutions are provided by relatively Python (e.g., *bcbio-nextgen* [21]) and Java packages (e.g., *bpipe* [22]). Other quite flexible solutions include *Snakemake* [23], *Nextflow* [24] and *BigDataScript* [25], which provide new domain-specific programming languages (DSL) to synthesize workflows and bioinformatics pipelines. Computational languages for the description and sharing of workflows, including *CWL* (common workflow language, software.broadinstitute.org/ wdl) and WDL (open workflow description language, www. commonwl.org) are also standard solutions.

B. *Collaboration Support Tools*

Collaborative support systems (CSS) relate to software designed to ease a group of people to elaborate on their common tasks and achieve their goals [26]. The range of CSS could be grouped into two main categories namely, *mind mapping* and *argumentation support* tools.

Mind Mapping. These tools allow the creation and processing of the so-called "*mind maps*". A mental map is a diagram formed to represent ideas, or other elements connected and arranged around a central topic to explore. Mental maps are mainly used to visualize and appropriately organize ideas offering problem-solving functionality to support decision-making. A mental map may be considered as an illustration of accumulated knowledge in the field. Around the central concept, usually in a circular layout and with lines connecting them to the central idea, other ideas and concepts can be added. Mind mapping tools are primarily designed to support brainstorming, mainly in environments with increased data volumes and multiple users. They provide appropriate notifications in order for a user to be aware of the changes made by other users to mental maps the creation of which the user has contributed. They also support detailed follow-up of the history of each mental map thus enabling the retrieval of any version of the mental map from its creation to its last version. Representative mind mapping tools include: *MindMeister.* (www.mindmeister.com), *Mindomo* (www. mindomo.com), *Bubbl.us* (www.bubbl.us), and *Xmind* (www. xmind.net).

Argumentation support. Argumentation is a verbal activity, which is usually conducted in plain language. A person involved in argumentation uses words and phrases to declare, ask or deny something, respond to someone else's statements, questions or denials, and so on [27]. Technologies supporting argumentative collaboration usually provide the means to structure and visualize discussions, exchange documents, and manage users. In addition, they aim to use the argument as a means to create a common basis among the engaged stakeholders to comprehend specific positions and arguments on a topic, to establish assumptions and criteria, and to build a collective consensus on it. Representative argumentation support tools include: *Araucaria* (araucaria.com puting.dundee.ac.uk/doku.php*)*, *DebateGraph* (debategraph.org), *Compendium* *(*compendium.open.ac.uk*)*, *CoPe_it!* (copeit.cti.gr), *Cohere (*cohere.open.ac.uk*)*, and *Dicode* (dicode-project.cti.gr)

C. *Insufficiencies of current workflow technology*

As we have mentioned already, Taverna and Galaxy are the most known bioinformatics SWf environments, while *myExperiment* is the broadly used repository for the Taverna workflows (www.myexperiment.org) [28]. Moreover, their popularity seems to decline, and other more flexible and open solutions, like Bioconductor (www.bioconductor.org) and Python-based workflows, start to take over. As a prove for this, we queried Google Scholar for articles related to the aforementioned environments, for a period of eleven years (2008 to 2018). Each query included the terms "bioinformatics" and "workflow" and "<environment>", with <environment> taking the values, "taverna", "galaxy", "myexperiment", "bioconductor" and "python" in each respective query. The hits of an environment for each year was normalized relatively to the sum of hits across all environments, getting a measure of *popularity trend* for each environment. The results are illustrated in Fig. 1.

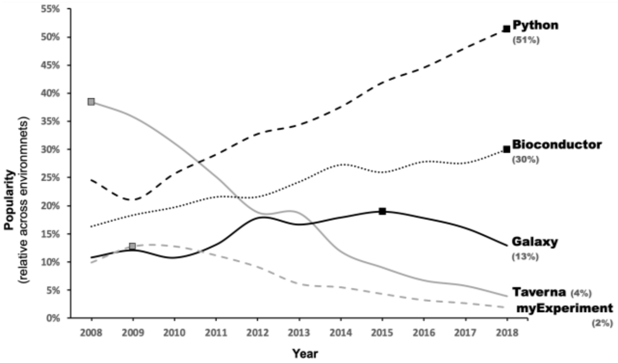

Figure 1. Popularity trends of established and widely utilized Bioinformatics workflow environments based on Google Scholar hits.

From Fig. 1, one could easily observe that the relative popularity of both Taverna and myExperiment constantly drops; for Taverna already from 2008 (denoted with the shaded square mark) to reach a relative popularity figure of 4% in 2018; and for myExperiment from 2009-2010 (where it enjoyed its maximum popularity) to reach a popularity figure of just 2% in 2018. Even Galaxy, the most recent and widely supported bioinformatics workflow environment, gained its maximum popularity around 2015 (~19%), and it then started to decline, reaching a relative popularity figure of 13% in 2018. Opposite trends are observed for Bioiconductor that started with a relative medium popularity in 2008 (~16%) and progressively reached a figure of 30%. The most noticeable observation concerns the popularity trend of Python-based workflows; it is constantly and aggressively increasing, reaching a relative popularity figure of 51% in nowadays. It seems that *one size does not fit all*, a fact that raises the challenge for open, flexible, customizable, and reproducibility supporting workflow environments [29].

III. Facing the Challenge

A. *The Need for Open-Science Methodologies*

Publishing the source code and making accessible the data of a research study allows the community not only to verify the reliability of the followed methodology but also, to use and adapt it to new research projects. Every new invention or tool evolves exactly through this process, i.e., *create* => *test* => *use* the tool, and it is strange why Bioinformatics did not embrace it, at least until the ´90s when initiatives such as the *Open Bioinformatics Foundation* (OBF, http://www.open-bio.org/wiki/BOSC) was formed, and the *BOSC* annual conferences began [30]. One of the OBF suggestions states that any scientific publication should incorporate and make available the whole research methodology [31]. This raises the need for the development of tools and services that enable the reproduction and verification of published results, a pretty complicated task, just because of the big number of available analytical tools. For example, Galaxy includes 3356 different tools, and myExperiment repository contains about 2700 workflows.

B. *Necessary Ingredients and Functionality*

Even if the concept of SWfs constitutes a relatively simple and old idea in the bioinformatics community, a number of critical but ignored factors constrain their wide adoption.

Embed and integrate. A SWf synthesis environment should provide mechanisms for the seamless integration of its constituents', without any "*selfish*" assumption that it is the sole and unique solution for the problem. With this assumption, any "*sovereignty*" behavior is avoided giving its place to "*contribution*" habits. Based on this, SWf environments should allow researchers to extract their full analysis in forms that can be easily assimilated by other contributions. *BASH* command scripts and scenarios (www.gnu.org/software/bash) with metadata described in known and easily manageable formats (e.g., *XML*, *JSON*), are typical examples that support such actions and research behaviors.

Integrate standard and validated bioinformatics tools. The inclusion of available, validated and widely used bio-data analysis tools in SWf environments provides a valuable development framework that can quickly and easily attract users to participate and contribute with their implementations. Such tools include genotype-phenotype association analysis (e.g., *plink*, *GATK*); genome annotation and functional prediction (e.g., *ANNOVAR*, *SIFT*, Polyphen-2); and next generation sequencing data analysis (e.g., *PRADA*, *Molgenis-inpute*).

Semantic annotation of resources. Most SWf environments ignore the semantic content of workflows and focus mainly on the analytical part of it. *Semantic enrichment* of SWfs brings a series of key-benefits to the engaged researchers including: search ordering and filtering; autocompletion (during query formulation); content grouping (according to users' interests and focus) and personalized recommendations [32]. Formation of the appropriate *meta-data* descriptions for the research objects included in a workflow, and their *ontology-based annotation* is the medium towards SWf semantic enrichment. For example, ontologies for gene function classification, like *Gene Ontology* [33]; for genetic diversity, like *VarioML* [34], as well for assessing and analyzing the evidence of target phenotypes, like *Human Phenotype Ontology* (HPO) [35], are extremely useful.

Virtualization. The computational execution environment of a SWf is often hidden to the user that compose and tries to run a workflow, and carries its own dependencies e.g., operating system, needed code libraries and packages. These requirements make it difficult to set-up a workflow execution environment, even for skilled IT personnel and experienced bioinformaticians. In addition, the lack of documentation makes the inexperienced users to mis-configured environments, and to the mismanagement of available resources. *Virtualization*, a "package in a package" solution for all the required components (software, operating system, libraries, and packages) seems as a promising approach [37]. The virtualized "*image*" or "*container*" may run on different operating systems, making virtualization a very convenient technique towards integrating workflows developed in different and heterogeneous environments. *Docker* (www.docker.com), perhaps the most widespread virtualization infrastructure, offers services to set-up and execute virtualized containers, and its value in contemporary data analytics has been already recognized [38], [39].

IV. THE OPENBIO-C PLATFORM

The OpenBio-C platform addresses the aforementioned challenges via the development of an integrated environment built around a number of interoperable components aiming to support the continuous development of reproducible bioinformatics workflows. The key-ingredients of the platform are detailed in the sequel.

A. The power of BASH

Nowadays, there are over a hundred tools for managing workflows. It is very common for each tool to define its own job description language (i.e., "Domain Specific Languages", DSL). Every new language, as simple as it may be, requires from users to devote time and effort to learn it. In addition, each language requires software that checks the syntactic and semantic correctness of the workflows described therein. The above adds unnecessary complexity to workflow systems and makes their spreading difficult. As an example, we quote that for 2018, the number of published jobs that used Galaxy was about 1,000 (galaxyproject.org/galaxy-project/statistics). If we assume that Galaxy is the most widespread bioinformatics workflow environment, and that more than 400,000 scientific works are published in the areas of biology, genetics and bioinformatics, we may conclude that the overwhelming majority of analyzes do not use a SWfMS. However, any task that requires the combination of more than one tool, or requires a tool that is only available in a Linux environment, then it utilizes the *BASH* environment. BASH is the most common command-line interface in Linux and OSX (there are also versions for Windows). Although BASH does not present a workflow synthesis environment, it is basically a framework where the individual steps of a workflow can be described. Also, most existing workflow management environments encode or describe the individual steps of a workflow in BASH format [40]. A basic innovation underlying OpenBio-C is that it *conceals* the user from the complexity and difficulty of BASH bringing it at the "surface" thus, making needless for the user to learn any new programming or specialized description language. Under these design principles: (i) flexibility in the design of workflows is strengthened; (ii) their customization is eased; and (iii) their sharing is achieved.

B. Importing tools and composing workflows

In the Bioinformatics scientific community there are tools considered as "basic" and tools that are more "experimental" (which are based on "basics" but rarely used). There are also common programming libraries that are used by a variety of other tools. This creates an *interdependence* between tools and libraries. An innovation of OpenBio-C is that the user can declare a tool as dependent on one or more other tools. These tools are independent research objects, exist in different records in the database, and have their own usage statistics and annotations. When a tool is fork(ed), it adopts the dependencies of the original tool (the user of course can change them). So, each tool is accompanied by its *dependency* tree, which is visible in the OpenBio-C graph-based workflow synthesis environment. To install a tool, the user simply inserts the BASH commands that install it.

C. The Execution Environment

When a user imports a tool, a dataset or a workflow, OpenBio-C tries to confirm their correctness. To do this, the BASH commands for all tools engaged with the current workflow are collected, and sent to an independent subsystem that has its own architecture (it runs in its own computing environment). This subsystem receives *requests* from OpenBio-C's back-end to run a BASH program. Upon receiving such a request, it activates a v*irtual computing environment* in which it executes the commands and when they terminate, it returns the execution results to the back-end. The virtual computing environment is configured through Docker. It is important to emphasize that the whole process is based on *asynchronous execution*. The back-end can send thousands of requests in a minute, which come in a queue of priorities. Subsequent processes then receive requests from the priority queue, execute them and return the result. This enables performing all requests regardless of the existing computing infrastructure. Following this mode of operation, the system remains "modular" and provides scaling by adding additional computing resources. Currently, its implementation is based on the Python 3 *asyncio* library (docs.python.org/3/ library/asyncio.html).

D. The Collaboration Component

A fundamental deficit of contemporary bioinformatics SWf environments is that they lack collaboration support services during the design of workflows. OpenBio-C tries to fill-in this gap by introducing and implementing an *argumentation-based collaboration component*. The collaboration component supports the following key features: (a) *Import - Edit - Delete* nodes in the discourse / argumentation graph; (b) *Edit* the text of a discourse block node; and (c) Provide brief information on the discussion topic through a specific *tooltip*. Every discussion is represented as an *argumentation graph* (Fig. 2). The *argumentation model* used is based on the *IBIS* (Issue-Based Information System) framework [41], following the design principles presented in [42]. Each node in the argumentation graph may belong to one of the following five categories / types: *issue* –a question / problem that a user has or a topic he/she posts for discussion; *solution* –a suggested solution to the issue under consideration; *position in favor* –an argument defending a proposed solution for the problem; *position aga*inst –a counter-argument to a proposed solution; *note* –a comment that does not affect the formal assessment of the discussion. Data from a conversation are stored in JSON files for further retrieval (i.e., recall a previous discussion).

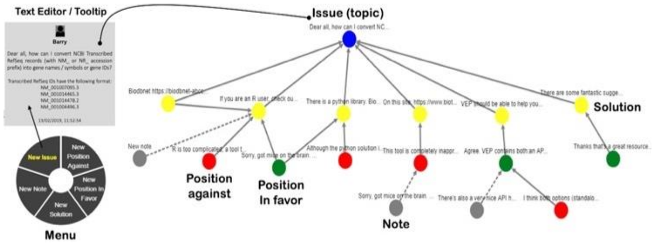
Text Editor / Tooltip
Barry
Dear all, how can I convert NCBI Transcribed RefSeq records (with NM_ or NR_ accession prefix) into gene names / symbols or gene IDs?
Transcribed RefSeq IDs have the following format:
NM_001007095.3
NM_001014465.3
NM_001014478.2
NM_001004496.3
13/02/2019, 11:52:54
Issue (topic)
Dear all, how can I convert NC...
Biodbnet https://biodbnet-abcc...
If you are an R user, check ou...
There is a python library. Bio...
On this site, https://www.biot...
VEP should be able to help you...
There are some fantastic sugge...
Solution
New note
R is too complicated, a tool t...
Sorry, got mice on the brain. ...
Although the python solution i...
This tool is completely inappr...
Agree. VEP contains both an AP...
Thanks that's a great resource...
New Issue
New Position Against
New Position In Favor
New Solution
New Note
Menu
Position against
Position In favor
Sorry, got mice on the brain. ...
There's also a very nice API h...
I think both options (standalo...
Note

Figure 2. The argumentation graph set-up of the collaboratrion component.

Fig. 2 depicts the basic graph-based set-up of an argumentation discourse, as supported by the collaboration component. It refers to a discussion carried out through *Reddit* (www.reddit.com), a widely used social news aggregation, web content rating, and discussion website. The specific post concerns a bioinformatics-related question posted by a user namely, "*How to convert NCBI RefSeq IDs to Gene names / symbols*". There were six solutions (answers) to the question, posted by other users (yellow nodes in Fig. 2). As expected, there are conflicts between solutions and different views. Specifically, there are arguments in favor of (green nodes) and against (red nodes) the six posted solutions. There is also the possibility to post some comments (gray nodes) linked with either the solutions or the arguments. Users of the collaboration component may create a new node in the argumentation graph by using the *menu* (Fig. 2, left-down part), shown by right-clicking in the collaboration workspace. The options available concern the creation of a new issue node, the creation of a new issue position in favor or against, the creation of a new solution node, and the creation of a new note node. When inserting a node, the creator and the date of creation are annotated and appropriately registered in the OpenBio-C database. The node creator has the ability to process / delete / add a description for the node through an embedded *text editor* (Fig. 2, top-left part).

The constituent components of OpenBio-C, and the data flow between them are illustrated in Fig. 3. The OpenBio-C platform can be launched via OpenBio-C's project website, (www.openbio.eu/platform).

E. *Working with OpenBio-C*

As a showcase we present a simple yet indicative example of how to add tools, compose workflows, execute them and monitor their execution through the OpenBio-C platform. We assume a user with an entry level of knowledge about bioinformatics, and interested in "*creating a PCA (Principal Component Analysis) plot of the world-wide genetic heterogeneity based on the HAPMAP3 data*". Initially the user posts a simple question on the Q&A section of OpenBio-C.

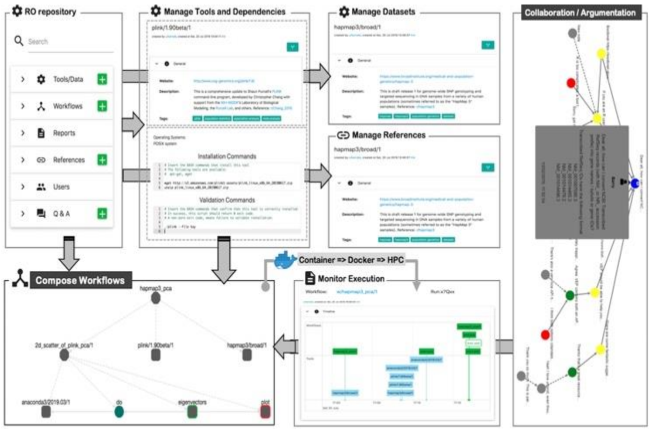

Figure 3. The components of the OpenBio-C SWf workflow environemnt.

Another user suggests that this task can be done with the *plink* tool (zzz.bwh.harvard.edu/plink). A third user with more advanced programming knowledge attempts to create the workflow in two simple steps: (i) *add* the tool by writing down the needed BASH *installation commands*, and *validate* the installation by inserting the BASH *installation commands* or, the commands that *confirm* successful installation. When trying to install a tool the system first executes the validation commands and if these run successfully then it considers that the tool is installed correctly. If the validation fails then the system runs the installation commands and then the validation commands again. If the validation fails again, an error message appears to the user; (ii) *set the engaged variables* by defining the engaged variables, the values of which are accessible by any other tool of the workflow that dependents from the one being added. Finally, the user saves the tool. Now that the tool is imported in the system, the user can create a workflow that performs the "PCA analysis". To do so, the user navigates in the "Workflows" section of OpenBio-C and creates a new workflow with the name "hapmap3_pca/1", by simply dragging and dropping the "plink/1.90beta/1" tool in the OpenBio-C *graph-based workflow editing window* (Fig. 4a), and adds the needed execution steps as well as other tools (e.g., "plot"). Finally, the whole workflow is *virtualized* using the Docker infrastructure (the specific image was run on a local cloud infrastructure). An important functionality of OpenBio-C is that it offers graph-based monitoring of the workflow execution (Fig. 4b).

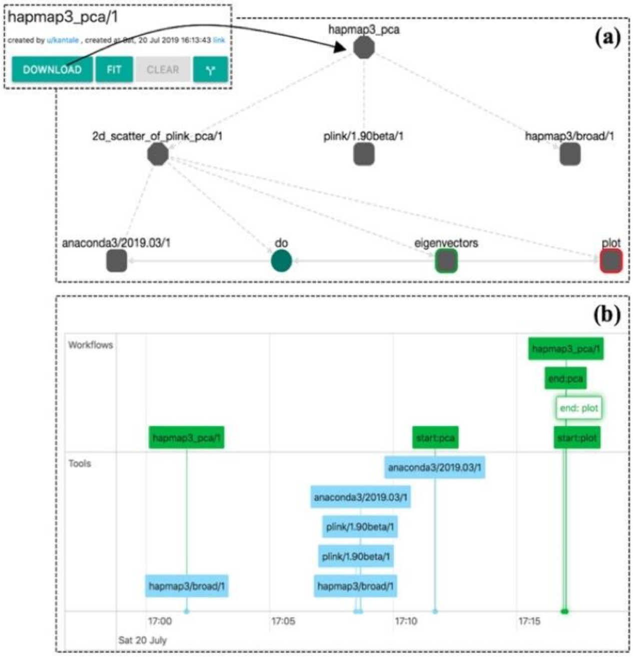

Figure 4. Graph-based workflow editing and monitoring in OpneBio-C.

ACKNOWLEDGMENT

The work presented in this paper is supported by the OpenBio-C project (www.openbio.eu) which is co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project id: T1EDK- 05275).

References

[1] L. Bornmann and R. Mutz, “Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references,” *J. Assoc. Inf. Sci. Technol.*, vol. 66, no. 11, pp. 2215–2222, 2015.

[2] J. A. Hill, “How to Review a Manuscript,” *J. Electrocardiol.*, vol. 49, no. 2, pp. 109–111, 2016.

[3] C. G. Begley and L. M. Ellis, “Raise standards for preclinical cancer research,” *Nature*, vol. 483, p. 531, Mar. 2012.

[4] F. Prinz, T. Schlange, and K. Asadullah, “Believe it or not: how much can we rely on published data on potential drug targets?,” *Nat. Rev. Drug Discov.*, vol. 10, p. 712, Aug. 2011.

[5] L. P. Freedman, I. M. Cockburn, and T. S. Simcoe, “The economics of reproducibility in preclinical research,” *PLOS Biol.*, vol. 13, no. 6, pp. 1–9, 2015.

[6] M. R. Munafò *et al.*, “A manifesto for reproducible science,” *Nat. Hum. Behav.*, vol. 1, p. 21, Jan. 2017.

[7] D. R. Grimes, C. T. Bauch, and J. P. A. Ioannidis, “Modelling science trustworthiness under publish or perish pressure,” *R. Soc. Open Sci.*, vol. 5, no. 1, p. 171511, Jul. 2019.

[8] Z. D. Stephens *et al.*, “Big data: Astronomical or genomical?,” *PLoS Biol.*, vol. 13, no. 7, 2015.

[9] K. Charlebois, N. Palmour, and B. M. Knoppers, “The Adoption of Cloud Computing in the Field of Genomics Research: The Influence of Ethical and Legal Issues,” *PLoS One*, vol. 11, no. 10, p. e0164347, Oct. 2016.

[10] V. Marx, “The big challenges of big data,” *Nature*, vol. 498, p. 255, Jun. 2013.

[11] T. Hulsen *et al.*, “From Big Data to Precision Medicine,” *Front. Med.*, vol. 6, p. 34, Mar. 2019.

[12] K. Y. He, D. Ge, and M. M. He, “Big Data Analytics for Genomic Medicine,” *Int. J. Mol. Sci.*, vol. 18, no. 2, p. 412, Feb. 2017.

[13] G. Poste, “Bring on the biomarkers,” *Nature*, vol. 469, no. 7329, pp. 156–157, 2011.

[14] L. A. Levin and H. V. Danesh-Meyer, “Lost in translation: Bumps in the road between bench and bedside,” *JAMA - Journal of the American Medical Association*, vol. 303, no. 15. pp. 1533–1534, 2010.

[15] J. Leipzig, “A review of bioinformatic pipeline frameworks,” *Brief. Bioinform.*, vol. 18, no. 3, pp. 530–536, May 2017.

[16] M. R. Karim, A. Michel, A. Zappa, P. Baranov, R. Sahay, and D. Rebholz-Schuhmann, “Improving data workflow systems with cloud services and use of open data for bioinformatics research,” *Brief. Bioinform.*, vol. 19, no. 5, pp. 1035–1050, Sep. 2018.

[17] O. Spjuth *et al.*, “Experiences with workflows for automating data-intensive bioinformatics,” *Biology Direct*, vol. 10, no. 1. 2015.

[18] N. Kulkarni *et al.*, “Reproducible bioinformatics project: a community for reproducible bioinformatics analysis pipelines,” *BMC Bioinformatics*, vol. 19, no. 10, p. 349, Oct. 2018.

[19] B. Giardine *et al.*, “Galaxy: A platform for interactive large-scale genome analysis,” *Genome Res.* , vol. 15, no. 10, pp. 1451–1455, Oct. 2005.

[20] K. Wolstencroft *et al.*, “The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud,” *Nucleic Acids Res.*, vol. 41, no. W1, pp. W557–W561, Jul. 2013.

[21] R. Guimera, “bcbio-nextgen: Automated, distributed next-gen sequencing pipeline,” *EMBnet.journal*, vol. 17, no. B, p. 30, 2012.

[22] S. P. Sadedin, B. Pope, and A. Oshlack, “Bpipe: a tool for running and managing bioinformatics pipelines,” *Bioinformatics*, vol. 28, no. 11, pp. 1525–1526, Apr. 2012.

[23] J. Köster and S. Rahmann, “Snakemake—a scalable bioinformatics workflow engine,” *Bioinformatics*, vol. 28, no. 19, pp. 2520–2522, Aug. 2012.

[24] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame, “Nextflow enables reproducible computational workflows,” *Nat. Biotechnol.*, vol. 35, no. 4, pp. 316–319, 2017.

[25] P. Cingolani, R. Sladek, and M. Blanchette, “BigDataScript: a scripting language for data pipelines,” *Bioinformatics*, vol. 31, no. 1, pp. 10–16, Jan. 2015.

[26] N. Karacapilidis, *Mastering Data-Intensive Collaboration and Decision Making: Research and Practical Applications in the Dicode Project*. Springer Publishing Company, Incorporated, 2014.

[27] F. H. Eemeren, R. Grootendorst, R. H. Johnson, C. Plantin, and C. Willard, *Fundamentals of argumentation theory: A handbook of historical backgrounds and contemporary developments*. Hillsdale, NJ, US: Lawrence Erlbaum Associates Inc, 1996.

[28] C. A. Goble *et al.*, “myExperiment: a repository and social network for the sharing of bioinformatics workflows,” *Nucleic Acids Res.*, vol. 38, no. Web Server issue, pp. W677–W682, Jul. 2010.

[29] Y. Zhang, B. Vasilescu, H. Wang, and V. Filkov, “One Size Does Not Fit All: An Empirical Study of Containerized Continuous Deployment Workflows,” in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018, pp. 295–306.

[30] N. L. Harris *et al.*, “The 2015 Bioinformatics Open Source Conference (BOSC 2015),” *PLOS Comput. Biol.*, vol. 12, no. 2, p. e1004691, Feb. 2016.

[31] K. Hettne *et al.*, “Workflow Forever: Semantic Web Semantic Models and Tools for Preserving and Digitally Publishing Computational Experiments,” in *Proceedings of the 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences*, 2012, pp. 36–37.

[32] J. Zarnegar, “Semantic enrichment of life sciences content: how it works and key benefits for researchers,” *Emerg. Top. Life Sci.*, vol. 2, no. 6, pp. 769 LP – 773, Dec. 2018.

[33] G. O. Consortium, “Gene Ontology Consortium: going forward,” *Nucleic Acids Res.*, vol. 43, no. Database issue, pp. D1049–D1056, Jan. 2015.

[34] M. Byrne *et al.*, “VarioML framework for comprehensive variation data representation and exchange,” *BMC Bioinformatics*, vol. 13, no. 1, p. 254, Oct. 2012.

[35] P. N. Robinson, S. Köhler, S. Bauer, D. Seelow, D. Horn, and S. Mundlos, “The Human Phenotype Ontology: a tool for annotating and analyzing human hereditary disease,” *Am. J. Hum. Genet.*, vol. 83, no. 5, pp. 610–615, Nov. 2008.

[36] J. Malone, R. Stevens, S. Jupp, T. Hancocks, H. Parkinson, and C. Brooksbank, “Ten Simple Rules for Selecting a Bio-ontology,” *PLoS Comput. Biol.*, vol. 12, no. 2, pp. e1004743–e1004743, Feb. 2016.

[37] G. Juve and E. Deelman, “Scientific Workflows in the Cloud - Grids, Clouds and Virtualization,” M. Cafaro and G. Aloisio, Eds. London: Springer London, 2011, pp. 71–91.

[38] C. Boettiger, “An introduction to Docker for reproducible research, with examples from the {R} environment,” *CoRR*, vol. abs/1410.0, 2014.

[39] P. Di Tommaso, E. Palumbo, M. Chatzou, P. Prieto, M. L. Heuer, and C. Notredame, “The impact of Docker containers on the performance of genomic pipelines,” *PeerJ*, vol. 3, pp. e1273–e1273, Sep. 2015.

[40] F. B. Menegidio *et al.*, “Bioportainer Workbench: a versatile and user-friendly system that integrates implementation, management, and use of bioinformatics resources in Docker environments,” *Gigascience*, vol. 8, no. 4, Apr. 2019.

[41] W. Kunz and H. W. J. Rittel, *Issues as Elements of Information Systems*, no. 131. Institute of Urban and Regional Development, University of California, 1970.

[42] O. Scheuer, F. Loll, N. Pinkwart, and B. M. McLaren, “Computer-supported argumentation: A review of the state of the art,” *Int. J. Comput. Collab. Learn.*, vol. 5, no. 1, pp. 43–102, 2010.