Integration techniques for modern bioinformatics workflows
Kanterakis, Alexandros

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2018

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.
Creating transparent and reproducible pipelines: Best practices for tools, data and workflow management systems

Alexandros Kanterakis\textsuperscript{1}, George Potamias\textsuperscript{2}, Morris A. Swertz\textsuperscript{1}

\textsuperscript{1}Genomics Coordination Center, Department of Genetics, University Medical Center Groningen, University of Groningen, Groningen, The Netherlands.
\textsuperscript{2}Institute of Computer Science, Foundation for Research and Technology Hellas (FORTH), Heraklion, Greece.

Submitted as a chapter to \textit{Human Genome Informatics: Translating Genes Into Health}
Editors: Darrol Baker, Christophe Lambert and George Patrinos
Publication August 2018
Publisher: Elsevier
https://www.elsevier.com/books/human-genome-informatics/baker/978-0-12-809414-3
Abstract

Today, the practice of properly sharing the source code, analysis pipelines and protocols of published studies has become commonplace in bioinformatics. Additionally, there is a plethora of technically mature Workflow Management Systems (WMS) that offer simple and user-friendly environments where users can submit tools and build transparent, shareable and reproducible pipelines. Arguably, the adoption of open science policies and the availability of efficient WMSs constitutes major progress towards battling the replication crisis, advancing research dissemination and creating new collaborations. Yet, today we still see that it is very difficult to include a large range of tools in a scientific pipeline, whereas on the other side, certain technical and design choices of modern WMSs discourage users from doing just this. Here we present three sets of easily applicable “best practices” targeting (i) bioinformatics tool developers, (ii) data curators and (iii) WMS engineers, respectively. These practices aim to make it easier to add tools to a pipeline, to make it easier to directly process data, and to make WMSs widely hospitable for any external tool or pipeline. We also show how following these guidelines can directly benefit the research community.

2.1 Introduction

Today, publishing the source code, data and other implementation details of a research project serves two basic purposes. The first is to allow the community to scrutinize, validate and confirm the correctness and soundness of a research methodology. The second is to allow the community to properly utilize the methodology in novel data, or to adjust it to test new research hypotheses. This is a natural process that pertains to practically every new invention or tool and can be reduced down to the oversimplified sequence: Create a tool, test the tool, use the tool. Yet it is surprising that in bioinformatics this natural sequence was not standard practice until the 1990s when specific groups and initiatives like BOSC [63] started advocating its use. Fortunately, today we can be assured that this process has become common practice, although there are still grounds for improvement. Many researchers state that ideally, the ‘materials and methods’ part of a scientific report should be an object that encapsulates the complete methodology, is directly available and executable, and should accompany (rather than supplement) any biologic investigation [69]. This highlights the need for another set of tools that automate the sharing, replication and validation of results, and the conclusions from published studies. Additionally, there is a need to include external tools and data easily [58] and to be able to generate more complex or integrated
research pipelines. This new family of tools is referred to as Workflow Management Systems (WMS) or Data Workflow Systems [79].

The tight integration of open source policies and WMSs is a highly anticipated milestone that promises to advance many aspects of bioinformatics. In section 2.6 we present some of these aspects, with fighting the replication crisis and advancing healthcare through clinical genetics being the most prominent. One critical question is where do we stand now on our way to making this milestone a reality. Estimating the number of tools that are part of reproducible analysis pipelines is not a trivial task. Galaxy [55] has a publicly available repository of available tools called ‘toolshed’ 1 which lists 3,356 different tools. myExperiment [56] is a social website where researchers can share Research Objects such as scientific workflows; it contains approximately 2,700 workflows. For comparison bio.tools [74], a community-based effort to list and document bioinformatics resources, lists 6,842 items2. Bioinformatics.ca [16] curates a list of 1,548 tools and 641 databases, OMICtools [68] contains approximately 5,000 tools and databases, and the Nucleic Acids Research journal curates a list of 1,700 databases [53]. Finally, it is estimated there are approximately 100 different WMSs 3 with very diverse design principles.

Despite the plethora of available repositories, tools, databases and WMSs, the task of combining multiple tools in a research pipeline is still considered a cumbersome procedure that requires above-average IT skills [86]. This task becomes even more difficult when the aim is to combine multiple existing pipelines, use more than one WMS, or to submit the analysis to a highly customized, High Performance Computing (HPC) environment. Since the progress and innovation in bioinformatics lies to a great extent in the correct combination of existing solutions, we should expect significant progress on the automation of pipeline building in the future[111]. In the meantime, today’s developers of bioinformatics tools and services can follow certain software development guidelines that will help future researchers tremendously in building complex pipelines with these tools. Additionally, data providers and curators can follow clear directions in order to improve the availability, re-usability and semantic description of these resources. Finally, WMS engineers should follow certain guidelines that can augment the inclusiveness, expressiveness and user-friendliness of these environments. Here we present a set of easy-to-follow guidelines for each of these groups.

1Galaxy Tool Shed: https://toolshed.g2.bx.psu.edu/
2https://bio.tools/stats
3Awesome Pipeline: https://github.com/pditemmaso/awesome-pipeline
2.2 Existing workflow environment

Thorough reviews on existing workflow environments in bioinformatics can be found in many relevant studies [86], [79], [130]. Nevertheless, it is worthwhile to take a brief look at some of the most well-known environments and frequently used techniques.

The most prominent workflow environment and perhaps the only success story in the area - with more than a decade of continuous development and use - is Galaxy [55]. Galaxy has managed to build a lively community that includes biologists and IT-experts. It also acts as an analysis frontend in many research institutes. Other features that have contributed significantly to Galaxy’s success are: (1) capability to build independent repositories of workflows and tools and allow the inclusion of these from one installation to another, (2) having a very basic and simple wrapping mechanism for ambiguous tools, (3) native support for a large set of High Performance Environments (HPC) like (TORQUE, cloud, grid) [82], and (4) offering a simple, interactive web-based environment for graph-like workflow creation. Despite Galaxy’s success, it still lacks many of the qualitative criteria that we will present later. Perhaps most important is the final criterion that discusses enabling users to collaborate in analyses, share, discuss and rate results, exchange ideas, and co-author scientific reports.

The second most well-known environment is perhaps TAVERNA [150]. TAVERNA is built around the integration of many web services and has limited adoption in the area of genetics. The reason for this is that it is a more complex framework, it is not a web environment and forces users to adhere to specific standards (i.e. BioMoby, WSDL). We believe that the reasons why TAVERNA is lagging GALAXY, despite having a professional development team and extensive funding, should be studied more deeply in order to generate valuable lessons for future reference.

In this thesis we use a new workflow environment, MOLGENIS-compute [21], [19], [20]. This environment offers tool integration and workflow description that is even simpler than GALAXY and it is built around MOLGENIS, another successful tool used in the field [131]. Moreover it comes with “batteries included” tools and workflows for genotype imputation and RNA-sequencing analysis.

Other environments with promising features include Omics Pipe [50] and EDGE [89] for next generation sequencing data, and Chipster [77] for microarray data. All these environments claim to be decentralized and community-based although their wide adoption by the community still needs to be proven.

Besides integrated environments, it is worth noting some workflow solutions at the level of programming languages for example, Python packages like Luigi and bebio-
nextgen [61], and Java packages like bpipe [118]. Other solutions are Snakemake⁴, Nextflow⁵ and BigDataScript [29], which are novel programming languages dedicated solely to scientific pipelines⁶. Describing a workflow in these packages gives some researchers the ability to execute their analyses easily in a plethora of environments, but unfortunately the packages are targeted at skilled IT users.

Finally, existing workflows and languages for workflow description and exchange are YAWL [140], Common Workflow Language⁷ and Workflow Description Language⁸. The support of one or more workflow languages by an environment is essential and comprises one of the most important qualitative criteria.

2.3 What software should be part of a scientific workflow?

Even in the early phases of development of a bioinformatics tool, one should take into consideration that it will become part of a yet-to-be-designed workflow at some point. Since the needs of that workflow are unknown at the moment, all the design principles used for the tool should focus on making it easy to configure, adapt and extend. This is on top of other quality criteria that bioinformatics methods should possess [149], [87]. After combining tens of different tools together for the scope of this thesis, we can present a checklist of qualitative criteria for modern bioinformatics tools to cover coding, quality checks, project organization and documentation.

• On coding:
  – Is the code readable or else written in an extrovert way, assuming that there is a community ready to read, correct and extend it?
  – Can people, unfamiliar with the project’s codebase get a good grasp of the architecture and module organization of the project?
  – Is the code written according to the idiom used by the majority of programmers of the chosen language (i.e. camel case for Java or underscores for Python?)
  – Is the code style (i.e. formatting, indentation, comment style) consistent throughout the project?

⁴Similar to ‘make’ tool, with a targeted repository of workflows https://bitbucket.org/snakemake/snakemake/wiki/Home
⁵http://www.nextflow.io/
⁶For more: https://www.biostars.org/p/91301/
⁷https://github.com/common-workflow-language/common-workflow-language
⁸https://github.com/broadinstitute/wdl
• On quality checks:
  – Do you provide test cases and unit tests that cover the entirety of the code base?
  – When you correct bugs, do you make the previously failing input a new test case?
  – Do you use assertions?
  – Does the test data come from “a real world problem”?
  – Does your tool generate logs that could easily trace errors on: input parameters, input data, user’s actions or implementation?

• On project organization
  – Is there a build tool that automates compilation, setup and installation?
  – Does the build tool check for necessary dependent libraries and software?
  – Have you tested the build tool in some basic commonly used environments?
  – Is the code hosted in an open repository? (i.e. github, bitbucket, gitlab)
  – Do you make incremental changes, with sufficient commit messages?
  – Do you “re-invent the wheel”? Or else, is any part of the project already implemented in a mature and easily embedded way that you don’t use?

• On Documentation
  – Do you describe the tool sufficiently well?
  – Is some part of the text more targeted to novice users?
  – Do you provide tutorials or step-by-step guides?
  – Do you document interfaces, basic logic and project structure?
  – Do you document installation, setup and configuration?
  – Do you document memory, CPU, disk space or bandwidth requirements?
  – Do you provide execution time for average use?
  – Is the documentation self-generated? (i.e. Sphinx)
  – Do you provide means of providing feedback or contacting the main developer(s)?
• Having a user interface is always a nice choice, but does it always support command line execution? Command line tools are far more easily adapted to pipelines.

• Recommendations for command line tools in bioinformatics include [125]: always have help screens, use stdout, stdin and stderr when applicable, check for as many errors as possible and raise exceptions when they occur, validate parameters, and do not hard code paths.

• Create a script (e.g. with BASH) or a program that takes care of the installation of the tool in a number of known operating systems.

• Adopt open and widely used standards and formats for input and output data.

• If the tool will most likely be at the end of an analysis pipeline and will create a large list of findings (e.g. thousands of variants) that require human inspection, consider generating an online database view. Excellent choices for this are MOLGENIS [131] and BioMart. These systems are easily installed and populated with data whilst they allow other researchers to explore the data with intuitive user interfaces.

• Finally choose light web interfaces rather than local GUIs. Web interfaces allow easy automatic interaction in contrast to local GUIs. Each tool can include a modular web interface that in turn can be a component of a larger web-site. Web frameworks like Django offer very simple solutions for this. An example of an integrated environment in Django is given by Cobb [30]. Tools that include web interfaces are Mutalyzer [148] and MutationInfo9.

Before checking these lists a novice bioinformatician might wonder what minimum IT knowledge is required in the area of bioinformatics. Apart from basic abstract notions in computer programming, newcomers should get experience in BASH scripting, the Linux operating system and modern interpreted languages like Python [110]. They should also get accustomed to working with basic software development techniques [100] and tools that promote openness and reproducibility [73], [90] like GIT [114].

These guidelines are not only targeted to novices but also to experienced users. The bio-star project was formed to counteract the complexity, bad quality, poor documentation and organization of bioinformatics software, which was discouraging potential new contributors. The purpose of this project was to create “low barrier entry” environments.

that enforce stability and scale-up community creation. Moreover, even if some of these guidelines are violated, publishing the code is still a very good idea [8]. It is important to note here that at one point the community was calling for early open publication of code, regardless of the quality, while at the same time part of the community was being judgmental and rejecting this step [35]. So having guidelines for qualitative software shouldn’t mean that the community does not support users who do not follow them mainly due to inexperience.

Another question is whether there should be guidelines for good practices on scientific software usage (apart from creating new software). Nevertheless, since software is actually an experimental tool in a scientific experiment, the abuse of the tool might actually be a good idea! For this reason the only guideline that is crucial when using scientific software regards tracking results and data provenance. This guideline urges researchers to be extremely cautious and responsible when monitoring, tracking and managing scientific data and logs that are generated from scientific software. In particular, software logs are no difference to wet-lab notebook logs and should always be kept (and archived) for further validation and confirmation\(^\text{10}\). Here we argue that this responsibility is so essential to the scientific ethic that it should not be delegated lightheartedly to automatic workflow systems without careful human inspection.

### 2.4 Preparing data for automatic workflow analysis

Scientific analysis workflows are autonomous research objects in the same fashion as independent tools and data. This means that workflows need to be decoupled from specific data and should be able to analyze data from sources beyond the reach of the bioinformatician author. Being able to locate open, self-described and directly accessible data is one of the most sensitive areas of life sciences research. There are two reasons for this. The first covers law, ethical and privacy concerns regarding the individuals participating in a study. The second is the reluctance of researchers to release data that only they have the benefit of accessing, thus giving them an advantage in scientific exploitations. This line of thought has placed scientific data management as a secondary concern, often given low priority in the scope of large projects and consortia. It is interesting in this regard to take a look at the conservative views of the medical community on open data policies [91]. These views consider the latest trends for openness as a side-effect of the technology from which they should be protected rather than as a revolutionary chance for progress. Instead of choosing sides in this controversy, we argue that technology itself can be used to resolve the issue. This is

\(^{10}\text{http://www.nature.com/gim/journal/v18/n2/full/gim2015163a.html} \)
feasible by making data repositories capable of providing research data whilst protecting private sensitive information and ensuring proper citation.

Today it is evident that the only way to derive clinically applicable results from the analysis of genetic data is through the collective analysis of diverse datasets that are open not only for analysis, but also for quality control and validation scrutiny [96]. For this reason we believe that significant progress has to be made in order to establish the right legal frameworks that will empower political and decision-making boards to create mechanisms that enforce and control the adaptation of open data policies. Even then, the most difficult change remains the paradigm shift that is needed: from the deep-rooted philosophy of researchers who treat data as personal or institutional property towards more open policies [5]. Nevertheless, we are optimistic - some changes have already started to take place due to the open data movement. Here, we present a checklist for open data management guidelines within a research project [134].

- Make long-term data storage plans early on. Most research data will remain valuable even decades after publication.
- Release the consent forms, or other legal documents and policies under which the data were collected and released.
- Make sure that the data are discoverable. Direct links should not be more distant than 2 or 3 clicks away from relevant search in a search engine.
- Consider submitting the data to an open repository, e.g. Gene Expression Omnibus or the European Genome-phenome Archive (EGA).
- Provide meta-data. Try to adopt open and widely used ontologies and formats, and make the datasets self-explanatory. Make sure all data are described uniformly according to the meta-data and make the meta-data specifications available.
- Provide the “full story” of the data. Experimental protocols, equipment, pre-processing steps, data quality checks and so on.
- Provide links to software that has been used for pre-processing and links to tools that are directly applicable for downstream analysis.
- Make the data citable and suggest a proper citation for researchers use. Also show the discoveries already made using the data and suggest further experiments.
- Provide a manager/supervisor’s email and contact information for potential feedback from the community.
2.5 Quality criteria for modern workflow environments

Although the notion of a scientific workflow is a simple concept with a historic presence in the bioinformatics community there are many factors that affect their overall quality that are still being overlooked today. In this section we present some of these factors.

2.5.1 Being able to embed and to be embedded

Any workflow should have the ability to embed other workflows as components regardless of their in-between unfamiliarity. Modern workflow environments tend to suffer from the “lock-in” syndrome. Namely, they demand their users invest considerable time and resources to wrap an external component with the required meta-data, libraries and configuration files into a workflow. Workflows should be agnostic regarding the possible components supported and should provide efficient mechanisms to wrap and include them with minimum effort.

Similarly, a workflow environment should not assume that it will be the primary tool with which the researcher make the complete analysis. This behavior is selfish and reveals a desire to dominate a market rather than to contribute to a community. That being said, workflow systems should offer researchers the ability to export their complete analysis in formats that are easily digested by other systems. Examples include simple BASH scripts with meta-data described in well-known and easily parsed formats like XML, JSON or YAML. Another option is to directly export meta-types, scripts and analysis code in serialized objects like PICKLE and CAMEL\(^\text{11}\) that can be easily loaded as a whole from other tools.

2.5.2 Support ontologies

Workflow systems tend to focus more on the analysis part of the tool and neglect the semantic part. The semantic enrichment of a workflow can be achieved by adhering and conforming to the correct ontologies. Analysis pipelines that focus on high throughput genomics, like Next Generation Sequencing (NGS), or proteomics have indeed a limited need for semantic integration mainly because a large part of this research landscape is uncharted so far. Nevertheless, when a pipeline approaches findings closer to the clinical level, the semantic enrichment is necessary. At this level, the plethora of findings, the variety of research pipelines, and sometimes the discrepancies between conclusions, can create a bewildering terrain. Therefore, a semantic integration through ontologies can provide common ground for direct comparison of findings and methods. Thus

\(^{11}\text{http://eev.ee/blog/2015/10/15/dont-use-pickle-use-camel/}\)
ontologies, for example for biobanks [109], gene function [34], genetic variation [22] and phenotypes [117], can be extremely helpful. Of course, ontologies are not the panacea to all these problems since they have their own issues to be considered [94].

2.5.3 Support virtualization

The computing requirements of a bioinformatics workflow are often unknown to the workflow author or user but include the processing power, memory consumption and time required. Additionally, the underlying computing environment has its own requirements (e.g., the operating system, installed libraries and preinstalled packages). Since a workflow environment has its own requirements and dependencies, it is cumbersome for even skilled and well IT-trained bioinformaticians to set it up and configure. Consequently, a considerable amount of valuable research time is spent in configuring and setting up a pipeline or a workflow environment. Additionally, lack of documentation, IT inexperience and time pressure lead to misconfigured environments that in turn leads to waste of resources and may even produce erroneous results12. A solution for this can be virtualization. Virtualization is the “bundling” of all required software, operating system, libraries and packages into a unique object (usually a file) that is called an “image” or “container”. This image can be executed in almost all known operating systems with the help of specific software, making this technique a very nice approach for the “be embeddable” feature discussed above. Any workflow environment that can be virtualized is automatically easily embeddable in any other system by just including the created image.

Some nice examples include the I2B2 (Informatics for Integrating Biology and the Bedside) consortium13 which offers the complete framework in a VMware image [139]. Another is the transMART software that is offered in a VMware or VirtualBox container [4]. Docker is also an open-source project that offers virtualization, as well as an open repository where users can browse, download and execute a vast collection of community generated containers. Docker borrows concepts from the GIT version control system. Namely, users can download a container, apply changes and “commit” the changes to a new container. The potential value of Docker in science has already been discussed [12], [41]; for example BioShaDock is a Docker repository of tools that can be seamlessly executed in Galaxy [104]. Other bioinformatics initiatives that are based on Docker

---

13https://www.i2b2.org/
are the CAMI (Critical Assessment of Metagenomic Interpretation\textsuperscript{14}), nucleotid.es\textsuperscript{15} and bioboxes.org. All these initiatives offer a testbed for comparison and evaluation of existing techniques mainly for NGS tasks.

2.5.4 Offer easy access to commonly used datasets

Over the last years, an increasing number of large research projects have generated large volumes of qualitative data that are part of almost all bioinformatics analysis pipelines. So far although locating and collecting the data is straightforward, their volume and their dispersed descriptions makes their inclusion a difficult task. Modern workflow environments should offer automatic methods to collect and use them and examples of relevant datasets are ExAC \textsuperscript{32} with 60,000 exomes, 1000 Genomes Project \textsuperscript{1}, GoNL \textsuperscript{135}, ENCODE project \textsuperscript{33}, METABRIC dataset \textsuperscript{37} and data released from large consortia like GIANT\textsuperscript{16}. Another impediment is that different consortia release data with different consent forms, for example in the European Genome-phenome Archive (EGA) \textsuperscript{84} each provider has a different access policy. Filling and submitting these forms is another task that can be automated (with information provided by users). Another essential feature should be the automatic generation of artificial data whenever this is requested for testing and simulation purposes \textsuperscript{105}.

2.5.5 Support and standardize data visualization

A feature that is partially supported by existing workflow environments is the inherent support for data visualization. Over the course of genetics research, certain visualization methods have become standard and are easily interpreted and widely understood by the community. Workflow environments should not only support the creation and inclusion of these visualization techniques, but also suggest them whenever a pipeline is being written.

For example in Genome-Wide Analysis Studies (GWAS), Manhattan plots for detecting significance, Q-Q plots for detecting inflation and Principal Component Analysis plots for detecting population stratification or the inclusion of visualizations from tools that have become standard in certain analysis pipelines, like LocusView and Haploview plots for GWAS. The environment should also support visualization for research data like haplotypes \textsuperscript{75}, \textsuperscript{9}, reads from sequencing \textsuperscript{71}, NGS data \textsuperscript{126}, \textsuperscript{136} and biological networks \textsuperscript{103}.

\textsuperscript{14}http://cami-challenge.org/
\textsuperscript{15}Very interesting presentation on virtualization: http://nucleotid.es/blog/why-use-containers/
\textsuperscript{16}https://www.broadinstitute.org/collaboration/giant/index.php/Main_Page
As an implementation note, in the last few years we have seen a huge improvement in the ability of Internet browsers to visualize large and complex data. This was possible mainly due to the wide adoption and standardization of JavaScript. Today JavaScript can be used as a standalone genome browser (for example JBrowse [128] although UCSC also allows this functionality [115]) and genome viewer (e.g. pileup.js [142]). JavaScript has allowed the inclusion of interactive plots like those from RShiny\textsuperscript{17} and the creation of aesthetically and informative novel forms of visualization with libraries like D3\textsuperscript{18}. Today there are hundreds of very useful, context-specific minor online tools in the area of bioinformatics. The superior advantage of JavaScript is that it is already installed (and used daily) in almost all Internet browsers by default. We therefore strongly advocate workflow environments that support JavaScript data visualization.

### 2.5.6 Enable “batteries included” Workflow Environments

A workflow environment, no matter how advanced or rich-in-features, is of no use to the bioinformatics community if it does not offer a basic collection of widely adopted and ready-to-use tools and workflows. The inclusion of these, even in early releases, will offer a valuable testing framework and will quickly attract users to become a contributing community. These tools and workflows can be split into the following categories:

**Tools:**

- Tools that are essential and can be used for a variety of purposes such as plink [113] and GATK [39].
- Tools for basic text file manipulation (e.g. column extraction), format validation, conversion and quality control.
- Tools for basic analysis, e.g. for variant detection, sequence alignment, variant calling, Principal Component Analysis, eQTL analysis, phasing, imputation, association analysis and meta-analysis.
- Wrappers for online tools. Interestingly, a large part of modern analysis demands the use of tools that require interaction through a web browser. These tools either do not offer a local offline version or operate on large datasets that it is inconvenient to store locally. A modern workflow should include these tools via special methods that emulate and automate browser interaction. Examples of

\textsuperscript{17}http://shiny.rstudio.com/

\textsuperscript{18}https://d3js.org/
these tools are: for annotation (e.g. Avia [144], ANNOVAR [146], GEMINI [108]),
function analysis of variation (e.g. Polyphen-2 [2], SIFT, CONSURF [24]) and
gene-set analysis (e.g. WebGestalt [145]).

Workflows:

- Existing workflows, e.g. for high throughput sequencing [102], RNA-sequencing
data [38], [138], identification of de novo mutations [119], and genotype imputation
[78].

- Protocols for downstream analysis, e.g. for functional variation [85], finding of
novel regulatory elements [120], investigation of somatic mutations for cancer
research [28] and delivering clinically significant findings [152].

- Meta-workflows for comparison and benchmarking [7].

The environment should also provide basic indications of their performance and
system requirements.

2.5.7 Facilitate data integration, both for import and export

Today, despite years of developing of formats, ontologies and standards in bioinformatics,
there is still uncertainty over the correct or “right” standard when it comes to tool or
service interoperability [133]. To demonstrate this, we present a list of 7 commonly used
bioinformatics databases for various –omics datasets in Table 2.1. Each database is
qualitative, built and maintained by professionals, offer modern user interfaces and
has been a standard source for information for numerous research workflows. Yet each
database offers a different standard as a primary method for automated data exchange.
Some are using raw data in a specific format, some have enabled API access, and some
delegate the access to secondary tools (i.e. BioMart). As long as the genetics field
generates data with varying rates and complexity, we should not expect this situation
to change in the near future. For this reason, a workflow environment that needs to
include these databases must offer methods for accessing these sources tailored to the
specifics of each database. This underlines the need for “write once, use many” tools.
A user who generates a tool that automates the access to a database and includes it
in a workflow environment should also make it public for other users too, while the
environment should allow and promote this policy after some basic quality checks of
course. The conclusion is that an environment that promotes the inclusion of many
diverse tools, workflows and databases will also promote the creation of tools that
automate data access.
2.5 Quality criteria for modern workflow environments

<table>
<thead>
<tr>
<th>-omics</th>
<th>Title</th>
<th>Access</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutations</td>
<td>Locus Specific Mutation Database</td>
<td>Atom, VarioML, JSON</td>
<td>[51]</td>
</tr>
<tr>
<td>IncRNA</td>
<td>IncRNASNP: a database of SNPs in IncRNAs and their potential functions in human and mouse</td>
<td>CSV files</td>
<td>[57]</td>
</tr>
<tr>
<td>Phenotype</td>
<td>The NCBI dbGaP database of genotypes and phenotypes</td>
<td>FTP</td>
<td>[93]</td>
</tr>
<tr>
<td>Diseases</td>
<td>Online Mendelian Inheritance in Man (OMIM)</td>
<td>XML, JSON, JSONP</td>
<td>[62]</td>
</tr>
<tr>
<td>Variation</td>
<td>ClinVar: public archive of relationships among sequence variation and human phenotype</td>
<td>FTP, VCF</td>
<td>[83]</td>
</tr>
<tr>
<td>Annotation</td>
<td>GENCODE: The reference human genome annotation for the ENCODE Project</td>
<td>GTF, GFF3</td>
<td>[64]</td>
</tr>
<tr>
<td>Functional Annotation</td>
<td>A promoter-level mammalian expression atlas (FANTOM)</td>
<td>BioMart</td>
<td>[52]</td>
</tr>
</tbody>
</table>

Table 2.1: A list of popular databases for –omics data along with their major access methods and formats.
2.5.8 Offer gateways for High Performance Computing Environments

The computational resources for a large part of modern workflows make their execution prohibitive in a local commodity computer. Fortunately, we have experienced an explosion of HPC options in the area of –omics data [11]. These options can be split into two categories: the first refers to software abstractions that allow parallel execution in an HPC environment, and the second refers to existing HPC options.

Regarding the first category the options are:

- Map-reduce systems [155], e.g. GATK [99].
- Specific Map-reduce implementations e.g. Hadoop, for example for NGS [107] data.
- SPARK [147].
- Multithread or multi-processing techniques, tailored for High Performance Workstations, e.g. WiggleTools [153].

The second category includes the available HPC options:

- Computer clusters and computational grids, or else job-oriented approaches [21].
- Cloud, e.g. example for whole genome analysis [154].
- Single High Performance Workstations.
- Workstations with optimized hardware for specific tasks, e.g. GPUs [19].

A workflow environment should be compatible with tools that belong to the first category and offer execution options that belong to the second. This integration is perhaps the most difficult challenge for a workflow environment because each HPC environment has a completely different architecture and access policies. Access policies include free-for-academic-use through nation-wide cluster infrastructure like Surf for the Netherlands and EDET for Greece, and paid for corporate providers like Amazon EC2 or other bioinformatics tailored cloud solutions like sense [20] and arvados [21]. Existing environments that offer execution for a plethora of HPC options are Yabi [72] for analysis and BioPlanet [70] for benchmarking and comparison.

---

19 http://richardcaseyhpc.com/parallel2/
20 https://sense.io/
21 https://arvados.org/
Another major consideration is the data policy. Submitting data for analysis in an external environment often violates strict (and sometimes very old) institutional data management policies. For this reason some researchers suggest that for sensitive data we should “bring computation to the data” [54] and not the opposite [47].

2.5.9 Engage users in collaborative experimentation and scientific authoring

Existing workflow environments focus on automating the analysis part of the scientific process. We anticipate that the new generation of environments will also include the second and maybe more vital part of this process: active and fruitful engagement in a creative discourse that will enable the collaborative co-authoring of complete scientific reports, reaching perhaps the level of published papers. For these purposes, workflow environments should enable users to build communities, invite other users and easily share workflows and results. The system should include techniques to allow users to comment, discuss and rate not only workflows, but results, ideas and techniques. It should also let users include rich text formatting descriptions of all parts of the analysis and automatically generate publishable reports. Finally it should also give proper credits to each user in the respective parts of the analysis to which they have contributed [10]. By offering these features workflow environments will become the center of scientific exploration rather than just a useful analysis tool.

This emerging trend -qualitative authoring pooled effort of an engaged community- is called crowdsourcing. It has been used successfully to accomplish tasks that otherwise would have required enormous amounts of work by a small group of researchers for example the monitoring of infectious diseases [98], investigating novel drug-drug interactions [112] and creating valuable resources for personal genomics (e.g. OpenSNP [59] and SNPedia [23]). Of course crowdsourcing techniques can introduce a certain bias in the analysis [65] and they should be done with extreme caution especially in the area of personal genomics [36]. This means the workflow environment should be able to apply moderation and detect cases where bias arises from a polarized user’s contribution.

The most prominent tools for online interactive analysis are Jupyter[22] and R Markdown[23]. The first mainly targets the Python community and the second the R community. In these environments the complete analysis can be easily shared and reproduced. Additionally, collaborative online scientific authoring is possible with services like

Overleaf\textsuperscript{24} and the direct hosting of generated reports can take place in a GIT enabled repository for impartial analysis or in arXiv\textsuperscript{25} for close-to-publication reports. Hence an ideal workflow environment could integrate these techniques and enrich them with user feedback via comments and ratings. Although this scenario sounds futuristic, it has long been envisioned by the open science movement.

2.6 Benefits from integrated workflow analysis in Bioinformatics

In this section we present the foreseeable short and long-term benefits for the bioinformatics community by following the criteria presented for workflow environments.

2.6.1 Enable meta-studies, combine datasets and increase statistical power.

One obvious benefit is to be able to perform integrated analysis and uncover previously unknown correlations or even causal mechanisms [92]. For example existing databases already include a huge amount if research data from GWAS that enable the meta-analysis and re-assessment of their findings [88], [76] and index them according to diseases and traits [18]. The improved statistical power can locate correlations for very diverse and environmentally affected phenotypes such as Body Mass Index [129] or even educational attainment [116] although the interpretation of socially sensitive findings should always be very cautious.

2.6.2 Include methods and data from other research disciplines

Automated analysis methods can generate the necessary abstraction layer, which will attract specialists from distant research areas with a minimum knowledge of genetics. For example, bioinformatics research can benefit greatly from “knowledge import” from areas like machine learning, data mining, medical informatics, medical sensors, artificial intelligence and supercomputing. The Encode project had a special “machine learning” track that generated very interesting findings [33]. Existing studies can also be enriched with data from social networks [31].

\textsuperscript{24}https://www.overleaf.com/
\textsuperscript{25}http://arxiv.org/
2.6.3 Fight the reproducibility crisis

The sparseness of data and analysis workflows from high-profile published studies is a contributing factor to what has been characterized as the reproducibility crisis in bioinformatics. Namely, researchers are failing to reproduce results from many prominent papers and some times-groundbreaking findings\textsuperscript{26}. The consequences of this can be really harsh, since many retractions occur that stigmatize the community in general [6], which in turn affect the overall financial viability of the field. One report stated, “..mistakes such as failure to reproduce the data, were followed by a 50–73% drop in NIH funding for related studies”\textsuperscript{27}.

Moreover, irreproducible science is of practically no use for the community nor for society and constitutes a waste of valuable funding [25]. Many researchers are blaming the publication process (which favors positive findings and original work rather that negative findings or confirmation of results) and are demanding either the adoption of open peer review policies [60], [13] or even the redesign of the complete publishing industry [45]. Yet, a more simple and applicable remedy for this crisis is the wide adoption (and maybe enforcement) of open workflow environments that contain the complete analyses of published papers.

2.6.4 Spread of open source policies in genetics and privacy protection

Today a significant part of NGS data analysis for research purposes is taking place with closed-source commercial or even academic software. The presence of this software in research workflows has been characterized as an impediment to science [143]. There is also an increase in commercial services that offer genetic data storage (e.g. Google\textsuperscript{28}) and analysis\textsuperscript{29}. Other worrisome trends are gene patents [97] and possible malevolent exploitation of easily de-anonymized traits and genomes that reside in public databases [3]. These trends raise ethical concerns and they also contribute to the reproducibility crises. Workflows that enable user collaboration can also monitor and guide the development of novel software from an initial draft to a mature and tested phase. This development can only happen of course in the open source domain. In the same way that GIT promoted open source policies by “socializing” software development, we believe that collaborative workflow environments can promote similar open source policies in genetics.

\textsuperscript{26}For a review of reports see: \url{http://simplystatistics.org/2013/12/16/a-summary-of-the-evidence-that-most-published-research-is-false/}
\textsuperscript{27}\url{http://blogs.nature.com/news/2012/11/retractions-stigmatize-scientific-fields-study-finds.html}
\textsuperscript{28}\url{https://cloud.google.com/genomics/}
\textsuperscript{29}For example: \url{https://insilicodb.com}
2.6.5 Help clinical genetics research

The clinical genetics field is tightly related to the availability of highly documented, well-tested and directly executable workflows. Even today there are genetic analysis workflows that deliver clinically applicable results in a few days or even hours. For example, complete whole genome sequencing pipeline for neonatal diagnosis [121] [26] or pathogen detection in intensive care units [106], where speed is of paramount importance. There are also clinical genetics workflows for the early diagnosis of Mendelian disorders [151], while the automation of genetic analysis benefit the early diagnosis of complex non-Mendelian rare [15] and common diseases like Alzheimer [43] and can simplify or replace complex procedures like amniocentesis for the prenatal diagnosis of Trisomy 21 [27]. Moreover cancer diagnosis and treatment has been revolutionized with the introduction of NGS data [101] and guidelines for clinical applications are already emerging [42], [127].

Workflows in clinical genetics do not have to be “disease specific”. Although still in its infancy, the idea of the universal inclusion of genetics in modern health care systems (called personalized or precision medicine) has been envisioned [66]. Probably the most prominent public figure in this area, Eric Topol, has emphasized the importance of DNA and RNA level data on individualized medicine throughout the lifespan of an individual [137]. This will allow the delivery of personalized drugs [49] and therapies [124] and will remedy current trends that threaten the survival of modern health care systems. The introduction of genetic screening as a routine test in healthcare, will guide medical practitioners (MPs) to take more informed decisions. Also, MPs will have to adjust phenotyping in order to fully exploit the potentials of genetic screening [67]. Therefore we expect that clinical genetics have the potential to fundamentally change the discipline of medicine.

These visionary and futuristic ideas of course entail many challenges and considerations [81]. So far, the most important are [132] the conflicting models [46], the lack of standards, and the uncertainty regarding the safety, effectiveness [40] and cost [44] of the clinical genetics guidelines generated. This is despite the initial promising results from GWAS [95] and much effort to create guidelines and mechanisms for reporting Clinically Significantly Findings (CFS) to cohort or health care participants [17], [80].

To tackle this, we suggest that modern workflow environments enabling multi-level

---


31 “An astonishing 86% of all full-time employees in the U.S. are now either overweight or suffer from a chronic (but often preventable) disease”, “the average cost of a family health insurance premium will surpass average household income by 2033”: [http://www.forbes.com/sites/forbesleadershipforum/2012/06/01/its-time-to-bet-on-genomics/](http://www.forbes.com/sites/forbesleadershipforum/2012/06/01/its-time-to-bet-on-genomics/)
data integration are the key to a solution. Adopting open data policies will allow large population level studies and meta-studies that will resolve existing discrepancies. A necessary prerequisite for this is also the inclusion of data from electronic Health Records [122] and BioBanks [141] (one successful initiative is the Dutch Lifelines cohort [123]). As long as findings from genetics research remains institutionally isolated and disconnected from medical records the realization of clinical genetics and personalized medicine will remain only as vision.

2.7 Discussion

Today we have an abundance of high quality data that describe nearly all stages of genetic regulation and transcription for various cell types, diseases and populations. DNA sequences, RNA sequences and expression, protein sequence and structure, small molecule metabolite structure are all routinely measured in unprecedented rates [48]. Moreover, the prospects in the area of data collection seem very bright as the throughput rates are increasing, the quality metrics are improving and the prices of obtaining such data are dropping. This leads us to the conclusion that the intermediate step that is missing between enormous data-collections and mechanistic or causal models is the data analysis. The recommended way to proceed should be via data-driven approaches. This in turn leads to a new question: How can we bring together diverse analysis methods in order to qualitatively process heterogeneous big data in a constantly changing technological landscape? This challenge requires the collective effort of the bioinformatics community and the effort can only flourish if development is focused on building re-usable components rather than isolated solutions.

Acknowledgments

Funding: The research leading to these results has received funding from the Ubbo Emmius Fund to AK and BBMRI-NL, a research infrastructure financed by the Netherlands Organization for Scientific Research (NWO project 184.021.007), to MS.
Bibliography


tool=pmcentrezrendertype=abstract.


Belinda Giardine, Cathy Riemer, Ross C Hardison, Richard Burhans, and El


[69] Kristina Hettne, Stian Soiland-Reyes, Graham Klyne, Khalid Belhajjame, Matthew Gamble, Sean Bechhofer, Marco Roos, and Oscar Corcho. Workflow


Michael T. Krieger, Oscar Torreno, Oswaldo Trelles, and Dieter Kranzlmüller. Building an open source cloud environment with auto-scaling resources for


[89] Po-E Li, Chien-Chi Lo, Joseph J. Anderson, Karen W. Davenport, Kimberly A. Bishop-Lilly, Yan Xu, Sanaa Ahmed, Shihai Feng, Vishwesh P. Mokashi, and


[104] François Moreews, Olivier Sallou, Yvan Le Bras, Grosjean Marie, Cyril Monjeaud, Thomas A Darde, Olivier Collin, and Christophe Blanchet. A curated Domain centric shared Docker registry linked to the Galaxy toolshed, jul 2015. URL https://hal.archives-ouvertes.fr/hal-01160514/.


[112] Alexander Pfundner, Tobias Schönberg, John Horn, Richard D Boyce, and Matthias Samwald. Utilizing the wikidata system to improve the quality of


[141] Gert-Jan B van Ommen, Outi Törnwall, Christian Bréchot, and Dagher et al. BBMRI-ERIC as a resource for pharmaceutical and life science industries: the


