Scientific workflow managers in metabolomics: an overview

Verhoeven, A.; Giera, M.; Mayboroda, O.A.

Analyst

ROYAL SOCIETY OF CHEMISTRY

CRITICAL REVIEW

View Article Online
View Journal | View Issue

Cite this: *Analyst*, 2020, **145**, 3801

Scientific workflow managers in metabolomics: an overview

Aswin Verhoeven, * Martin Giera and Oleg A. Mayboroda

Received 7th February 2020,
Accepted 28th April 2020

DOI: 10.1039/d0an00272k

rsc.li/analyst

Providing maximum information on the provenance of scientific results in life sciences is getting considerable attention since the widely publicized reproducibility crisis. Improving the reproducibility of data processing and analysis workflows is part of this movement and may help achieve clinical deployment quicker. Scientific workflow managers can be valuable tools towards achieving this goal. Although these platforms are already well established in the field of genomics and other omics fields, in metabolomics scripts and dedicated software packages are still more popular. However, versatile workflows for metabolomics exist in the KNIME and Galaxy platforms. We will here summarize the available options of scientific workflow managers dedicated to metabolomics analysis.

Introduction

Advances in protein and genome sequencing and structure elucidation have exponentially increased the amount of data generated.[1,2] The human genome project was officially finished in 2003 (two years before schedule) and a draft of the human proteome published in 2014.[3,4] This burst of biological data coincided with the development of fast and cheap computer hardware, powerful high-level programming languages and comfortable software development environments. In turn, all this stimulated the rapid development of the field of bioinformatics. The contemporary standards of proteomics and genomics data analysis require a high degree of automation. Sophisticated software packages have been written to process and analyze raw experimental data quickly and reliably, exploiting the highly regular structure of the biological polymers (DNA, RNA and proteins). Yet, one very important level of biological regulation was not accommodated by these biopolymer analysis tools. Approximately 15% of the human protein coding genes (of those for which the function is known) code for products responsible for transformations of an extremely diverse and heterogeneous family of compounds that are commonly called metabolites. The pursuit of comprehensively quantifying the set of metabolites (the metabolome) present in specific parts of the organism or in specific biofluids is the field of metabolomics.[5,6] Clinical metabolomics shows great promise in the diagnostics of metabolic diseases,[7] cardiovascular diseases[8] and even all-cause mortality.[9]

Center for Proteomics and Metabolomics, Leiden University Medical Center, Albinusdreef 2, 2333ZA Leiden, The Netherlands. E-mail: A.Verhoeven@lumc.nl

The automatic processing of raw metabolomics data is far less advanced than that of most other omics fields. The chemical heterogeneity of metabolites is mainly to blame here. Chromatographic separation combined with mass spectrometry is the central technology in the field of proteomics and is the driving force behind its successes. This is also true for metabolomics, but the diversity of metabolites means that only a relatively small part of the metabolome can be reliably measured for a specific experimental setup. Only metabolites falling into a specific charge, mass, polarity and concentration range can be analyzed in one measurement. Thus, in contrast to proteomics and genomics, metabolomics from its beginning was never restricted to a single successful technology but used a wider range of analytical approaches. Besides LC-MS analysis, one of the most widely applied techniques for metabolomics analysis is nuclear magnetic resonance spectroscopy (NMR). The sensitivity of NMR is typically less than that of mass spectrometry, but NMR has a number of unique advantages, for instance the generally simple sample preparation methods, the non-destructive nature of the measurement, and the linear behaviour of NMR signals in response to concentration changes.[10]

In the field of metabolomics, and especially for NMR metabolomics, as much time and effort has been invested in the development of sample preparation methods and experimental techniques as in the processing and analysis of the data. Many articles have been published detailing new approaches, tools and algorithms that improve upon the various steps in the pipeline.[11] In the early days of metabolomics researchers often were limited to software supplied by the equipment manufacturer and general statistical software such as TopSpin,[12] Excel,[13] SPSS[14] and Simca.[15] These software packages contain convenient GUI interfaces and will be familiar to many scien-

tists. However, a big issue with these software packages is the repeatability of the data processing and analysis workflow. The workflow consists of numerous mouse operations and manually entered parameters that are hard or impossible to retrieve or reapply in an identical way. Importing the output of one tool into the software that handles the next step of the workflow is often a manual step and unless the user is exceedingly meticulous in his documentation it may not be clear how it was achieved. Typically, the workflow involves steps that can be carried out either manually or in an automated fashion. The reproducibility of the manual parts depends heavily on whether the software in which the steps are performed documents all the user's operations, and how well the user documents his or her own actions. Lacking documentation affects the reproducibility of the data processing and analysis. According to Goodman *et al.*,[16] there are three distinct forms of reproducibility: methods reproducibility (the ability to implement the experimental and computational methods), results reproducibility (the ability to replicate the results) and inferential reproducibility (the ability to draw identical conclusions). Being able to inspect and repeat the data processing and analysis part of a study clearly concerns the methods reproducibility. Methods reproducibility is one of multiple factors that contribute to the reproducibility crisis that currently affects the biomedical sciences.[17] Combining all steps into a single monolithic program dedicated to metabolomics circumvents this problem to some extent; examples of such software packages are Chenomx[18] for NMR and MAVEN[19] for LC-MS-based metabolomics studies. However, as soon as the user wants to add new tools or new technologies to the workflow, the problem returns.

Metabolite diversity is also the reason that both in mass spectrometry and in NMR peak assignment can sometimes be quite difficult. A single LC-MS run or 1D NMR spectrum is often not enough to assign peaks with 100% certainty. These analyses can then be complemented by MS/MS and 2D NMR experiments, but in both cases these experiments are more complex, more time-consuming and less sensitive than the simpler experiments. The difficulty with peak assignment and quantification led many researchers to avoid this altogether as a first step and use the raw spectra for constructing statistical models that relate the metabolome to a specific phenotype. This approach, also called untargeted metabolomics, was often the only approach that was used in earlier metabolomics projects. In theory, a machine learning model that successfully distinguishes cases and controls based on the NMR/MS spectra of the metabolome may be useful even without information on exactly what metabolites drive the model. For example, a machine model that successfully distinguishes urinary tract infection from other ailments could be very useful in a clinical context. However, the equipment that is used for metabolomics is expensive both in purchase and in use, making it hard to apply in a clinical setting. Knowing which limited set of metabolites is associated with the phenotype in question allows for the design of simpler and cheaper methods that achieve the same result. Moreover, the identities of the associated metabolites and the fold changes of their concentrations could shine light on the biochemistry behind the phenotype.[20] Combining targeted and untargeted methods is therefore important. This generates multiple data pipelines that need to be properly handled and finally combined.

To bring the field of metabolomics to the same level as its fellow -omics fields, and help solve the reproducibility crisis, what is needed is software that is capable of managing the entire data processing pipeline all the way from the raw data to publishable results. A software platform that achieves this is called a scientific workflow system. This review will look at the available scientific workflow systems for metabolomics and how well they succeed in contributing to reproducible research, among other features. Solutions for scientific data management that satisfy the FAIR guiding principles[21] already exist in the MetaboLights[22] and Metabolomics Workbench[23] repositories; with the establishment of scientific workflow systems the data processing could be integrated with those services.

Traits of a scientific workflow system

Scientific workflow systems integrate multiple steps of a data processing and analysis pipeline, ideally in its entirety. Tools that perform only one or a few of these steps and are not flexible enough for the integration of additional steps cannot be regarded as workflow platforms. Instead, they are just the constituent parts that can potentially be assembled into a workflow.

Ideally, a workflow platform should stimulate and support the user in achieving the following goals:

1. It should as much as possible achieve methods reproducibility for the data processing and analysis. Both the user himself and other scientists should be able to inspect, retrace and rerun the data analysis process from start to finish even years after the project has ended. Intermediate results should be stored at appropriate points in the workflow to allow for this inspection. Ideally, the state of an executed workflow at every point should be preserved and remain consistent with the rest of the workflow.

2. Likewise, it should be easy to extend the workflow with new tools; this allows the time invested into developing the workflow to not go to waste while possibly extracting new insights from the data that were hidden before. New algorithms for spectral analysis and machine learning may be developed in the future. The workflow should be able to absorb these new methods without these being specifically designed for the workflow platform. Likewise, the current standard set of technologies may be extended in the future with new NMR pulse sequences or new mass spectrometry methods, or even totally new inventions such as metabolomics by electron diffraction measurements. Extending the workflow with these new technologies should be accessible to everyone, not just the prime developers of the workflow.

3. Related but not identical to the previous point is that a workflow should stimulate creativity. Each metabolomics project has its own peculiarities, and the exploratory data analysis of the data may profit by deviating from the standard workflow. A good workflow platform should easily allow this, and not try to force the user into a straitjacket. This can happen at any point in the workflow; the spectra may require an alternative processing method, the project may need an alternative normalization approach, or it may require an unconventional statistical analysis. The requirement for these changes may only become apparent when using the workflow, and the user should be able to adapt the workflow to his or her needs without leaving the workflow platform itself (which would harm reproducibility) or without begging the developers of the workflow platform. The platform should allow for a quick and easy edit-run-debug cycle, not to hamper this creativity.

4. A workflow should be accessible to both experts and novices, expert coders and non-programmers. Scientists with expertise in sample preparation, performing measurements and/or the biological background of the project should still be able to execute the workflow and study the results of their efforts. At the same time, the workflow should not limit power users.

5. A workflow should be scalable. Many metabolomics projects start small, with perhaps a few tens of samples as a pilot project. The results then inspire the collection of a larger cohort of a few hundred or thousand samples. If the effects are subtle but the implications important, it may be decided to assemble a cohort of more than ten thousand people. Ideally, it should be possible to recycle the pilot workflow for the analysis of the giant cohort, perhaps on a larger desktop computer or server. Alternatively, the processing can be transferred to the cloud, but this is only necessary in extreme cases; metabolomics data cannot yet be regarded as big data since it rarely comprises more than 20 GB,[24] although it should be noted that the computing power offered by a cloud environment can also benefit smaller datasets.

6. It should be easy to troubleshoot workflows that raise errors or produce nonsense results. A workflow in the hands of a new user or applied to a new project may end up in a state that was not envisioned or considered by the workflow designer. The workflow platform should allow the user to investigate and correct the problem himself.

7. The workflow should be easily archivable and shareable. The data and the algorithms should be tightly associated with each other, so that if the project is retrieved from the archive, reevaluating the workflow again should be trivial; neither the algorithms nor the data should get "lost" during the archiving and retrieval process. Likewise, it should be easy to hand over the workflow to a different user who may wish to check the workflow (for example, a supervisor checking the work of a student) or a user who wishes to continue the project of his predecessor; a workflow should allow this user to continue exactly where his predecessor left off. For this reason, it is desirable that the algorithms that constitute the workflow and the workflow management system itself are free open-source software so that there are no licensing problems when sharing a workflow outside an organization, extremely important in academia where research is often conducted in the context of international consortia.

Workflows and workflow systems used in metabolomics

Workflows can be assembled in two different ways: either by calling the tools from some type of scripting language, or by integrating them in a dependency graph (or DAG, directed acyclic graph). The prototypical scripting language is the Bourne shell that is common in the world of Unix-type operating systems. Today, however, these scripts are usually written in the R or Python languages.[25,26] The scientific notebook systems provided by RStudio and Jupyter enable a convenient combination of code with documentation and data visualizations. Data processing pipelines constructed with scripts are very flexible and the R and Python ecosystems provide a huge number of libraries covering a wide range of functionalities. This also allows for advanced collaboration and version management by using version control systems such as Git. It is possible to call R from Python or Python from R, combining the two ecosystems. However, while RStudio and Jupyter are very good development platforms, they do not manage data. The state of the system at a certain point in the workflow is not stable; if lines of code are executed out of sequence during the exploratory phase of the data analysis, the same line of code at the same point in the script does not always produce the same result. Also, it is easy for a script-based workflow to become obscured by the programming logic. This also limits the accessibility to people with little programming experience. Therefore, RStudio- and Jupyter-based platforms are development environments but not scientific workflow managers.

A web-based tool that has achieved a considerable level of popularity in the field of metabolomics is MetaboAnalyst.[27] It provides a convenient web-based interface to the most commonly used methods used for metabolomics analysis. One particularly nice feature is that it gives the equivalent R code for all the browser-operations, so that the analysis can be easily repeated in a properly configured R environment elsewhere. The MetaboAnalystR library[28] that these scripts rely on is free open-source software. But as such, MetaboAnalyst should be considered as a very user-friendly (but less flexible) version of an RStudio notebook and not really a workflow manager. MetaMS[29] is also an R library that aims to provide a full pipeline for LC-MS/GC-MS data processing and connects well with the MetaboLights repository, but lacks a convenient webinterface. Other web-based tools, such as Bayesil,[30] only live on the server of its designers and the reproducibility of the analysis depends on the entirely on the maintainers. Newer versions may behave differently from older ones, making it impossible the fully reproduce older results. In some cases the web-based tool may be entirely inaccessible.[31]

The alternative to a script is to define the data processing by a dependency graph of individual processing steps. In this context, a dependency graph defines for every step what other processing steps are required to generate its required input. This also typically includes a mechanism that stores intermediate results and that in the event of a configuration change in one step automatically and exclusively recalculates those steps that are dependent on it. One of the oldest and most basic tools to manage dependency graphs is the "make" utility, commonly used to call the compiler in a large software development project. "Make" speeds up software build times by only compiling those files of the software development project that have been edited since the previous build, and parts that are linked to the edited files. However, "make" is very general, and any set of tools that depend on each other's input and output can be handled by make. "Make" works by supplying it with a collection of individual dependencies in a so-called "makefile". From the "makefile" make internally builds the dependency graph. This graph is usually not visible to the user of make. This is fine for a software development project, but for a data processing workflow, a visual representation of the workflow is nearly essential. There are also dedicated scientific workflow tools such as Snakemake[32] that follow the make philosophy. Pegasus[33] also relies on DAGs but is designed in a way that that workflows can be easily moved between and optimized for various execution environments. Hyperflow[34] is a workflow system that allows JavaScript code to be integrated with the workflow implementation and is specifically aimed at experienced programmers. Nextflow[35] is another text-based tool, but it follows a more dataflow-like approach, in which multiple steps can execute simultaneously for increased performance and reduced storage space requirements. The dependency graph of these can be visualized, but not manipulated, by the Graphviz tool.[36] More domain-specific, Global Natural Products Social Molecular Networking (GNPS)[37] is an environment that, among other services, uses workflows based on the ProteoSAFe engine.[38] However, the workflow construction process is not well-documented.

Ideally, a workflow manager would allow the user to build, rearrange and extend the workflow dependency graph in a graphical environment. There are a number of workflow managers with a graphical user interface (GUI). Apache Taverna,[39] a Java-based workflow manager, is very good in combining various web services and has an automatic graph layout system. Orange,[40] a Python-based workflow manager/data mining platform, allows the user to assemble Python tools into graphical data flow graphs. Kepler[41] is a workflow system that works particularly well together with R and has a customized version specifically aimed at bioinformatics, BioKepler.[42] Rapidminer,[43] Pipeline Pilot[44] and Alteryx,[45] commercial options, are used a lot in industry. To our knowledge only two free open-source workflow managers have been used to build general metabolomics workflows. Those are Galaxy[46] and KNIME.[47,48]

Galaxy is a scientific workflow manager that operates as a webserver. Consequently, the UI is accessed through a web browser where the workflow graph and configuration UI elements are presented (see Fig. 1). The Galaxy engine can then be run locally or in the cloud. Galaxy basically operates as the "make" command with a GUI, calling R scripts, Python scripts and other command line tools using carefully configured recipes that describe the inputs and outputs of said commands. Adding new tools to the platform involves developing those tools in an external development environment. After the development is done and the tool works as desired, an XML file is written that provides the recipe on how the tool can be invoked from Galaxy. Tools provided by the community can be installed through the Galaxy Tool Shed.

The KNIME Analytics Platform, on the other hand, behaves more like a database, in which the workflow graph (Fig. 2) describes how data tables are filtered, transformed and otherwise manipulated. KNIME also provides various loop constructs, including recursive loops (and hence is not purely a DAG). Instead of only relying on external tools, KNIME provides hundreds of KNIME-specific nodes written in Java within the KNIME framework. R and Python scripts can be integrated

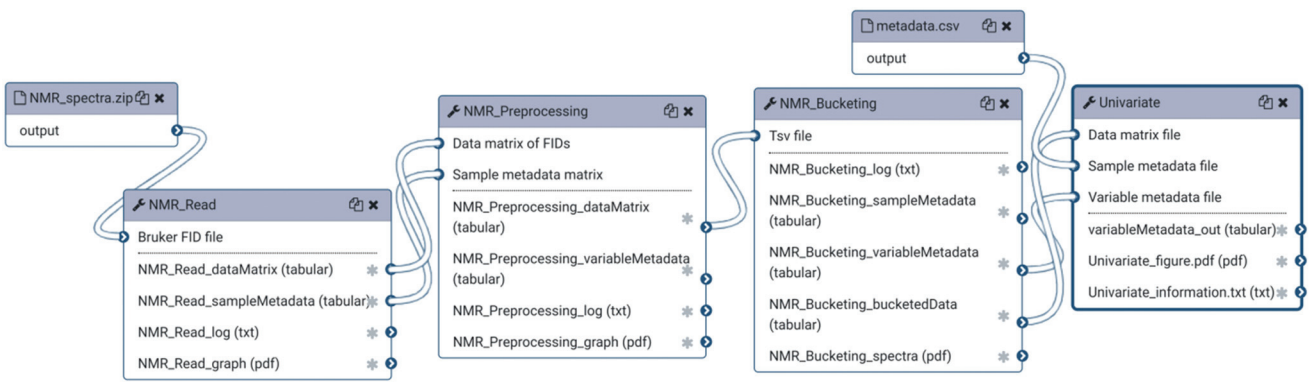

Fig. 1 An example of a Workflow4Metabolomics Galaxy workflow for NMR data. The workflow can either be assembled by dragging and connecting the tools, or by consecutively calling the tools on an imported dataset after which the workflow can be automatically generated from the data processing history.

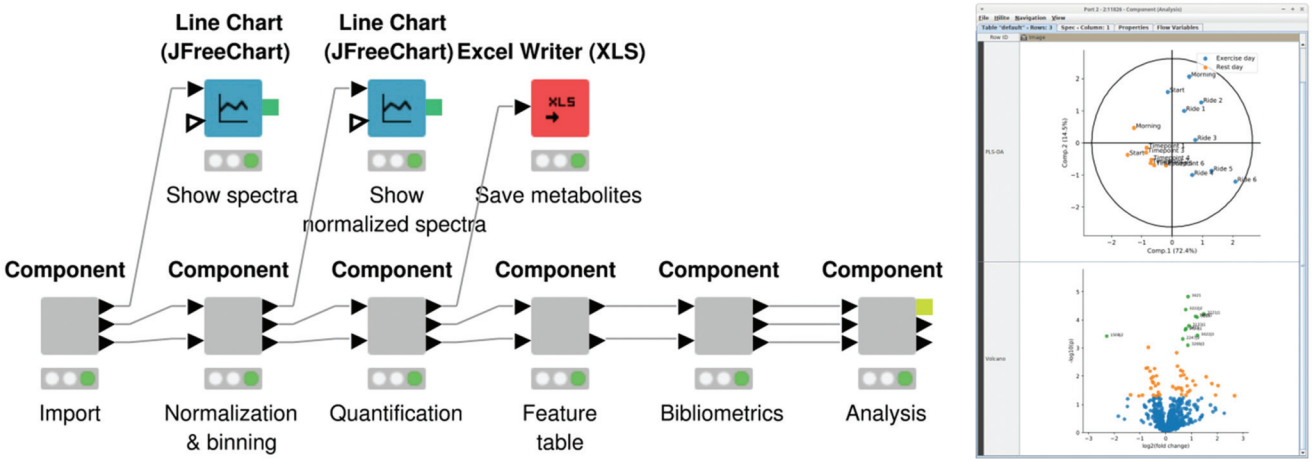

Fig. 2 This KIMBLE KNIME workflow shows native KNIME nodes for data plotting and export, and Components that are a workflow abstraction mechanism and contain more basic nodes. Green "traffic lights" indicate that output is available and can be inspected. The window shows the output of the second outport of the analysis node.

when they accept an R/Pandas dataframe as input and return the result as a new dataframe. These scripts can be developed in the KNIME workflow itself as KNIME provides small IDEs for those languages. Therefore, while Galaxy is a pure workflow manager, KNIME is both a workflow manager and a development platform. KNIME AG, the commercial entity behind the free open-source KNIME Analytics Platform, also produces the KNIME Server, through which workflows can be edited, configured and executed remotely or in the cloud. However, in contrast with the KNIME Analytics Platform, KNIME Server is a non-free product. It should be noted that both KNIME and Galaxy recently acquired the ability to edit and integrate Jupyter notebooks inside the workflow.[49,50]

Workflow4Metabolomics (W4M)[51] and PhenoMeNal[52] (which incorporates a large part of W4M) are metabolomics workflows built on top of the Galaxy platform. The organizations behind these workflows provide infrastructure to run the workflows in the cloud, but they can also be run from a local virtual machine or Docker container. Since Galaxy originates from the field of genomics, it is well suited to handle huge amounts of data in a reproducible manner. It is very well-suited for handling established workflows, *i.e.* workflows and toolsets that have achieved a certain level of maturity and general acceptance in the scientific community so that deviating from it is generally unnecessary or even strongly discouraged except for expert users. PhenoMeNal includes many established metabolomics tools for both NMR and MS-based metabolomics provided by a multitude of research groups. Furthermore, it provides facilities for distributed processing of workflows by managing multiple containers, leading to exceptional workflow scalability.[53] In general, Galaxy is very well-suited for handling containers, and an overview of this topic was given in a recent review.[54] Galaxy-M[55] is another Galaxy-based metabolomics workflow that relies on code compiled with the commercial Matlab environment.

KIMBLE[56] and KniMet[57] are based on the KNIME analytics platform, with KIMBLE focusing on NMR-based metabolomics and KniMet handling metabolomics data from MS measurements. A node library specifically for handling NMR data does not yet exist in KNIME, therefore KIMBLE uses Python script nodes that rely on the NMRGlue[58] library for handling NMR-specific processing. KNIME can automatically install KNIME node libraries required by the workflow, but it cannot do this with the Python environment and the necessary Python libraries. To get around this inconvenience, KIMBLE is provided as a virtual machine image that includes the KNIME Analytics platform, the Python environment and the Python libraries. Likewise, R and various R libraries relevant for metabolomics are also included in the virtual machine. This has the added benefit of improving reproducibility by making it possible to archive the software environment of the workflow.[59] The KniMet KNIME workflow can be extended with OpenMS[60] node library, a collection of KNIME nodes for handling mass spectrometry data. KIMBLE and KniMet can be easily combined by copying the workflows to the same workflow canvas.

How do Galaxy and KNIME compare with respect to the list of desirable workflow features?

1. Being workflow managers, both the Galaxy-based and KNIME-based workflows achieve data processing reproducibility by storing the workflow steps and the input and output data, as well as the intermediate results. Consistency is maintained in KNIME by resetting all subsequent nodes upon a configuration change, and in Galaxy by creating a new entry at the bottom of the history.

2. Both KNIME and Galaxy have repositories (KNIME has the KNIME Hub while Galaxy has the Galaxy Tool Shed) that can be used to extend existing workflows. KNIME allows the development of new KNIME-nodes, although this requires knowledge of the KNIME framework. Alternatively, KNIME's

capabilities can be extended by running Python or R scripts. The script code itself is embedded in the workflow. However, this does not apply to the libraries these scripts rely on. Galaxy is completely based on managing external tools. Galaxy also takes care of the dependencies of the tools in the Tool Shed or uses Conda for that.

3. The extent to which a workflow manager allows creativity is given by how quickly different processing and analysis methods can be tried out. In KNIME, new processing and analysis methods can be easily attempted by writing and executing R and Python scripts in KNIME nodes themselves. It also allows algorithms to be implemented with KNIME nodes itself, thanks to the availability of flow variables and various loop constructs. Loops are not available in Galaxy, there it is recommended to experiment with new analysis methods in external software such as RStudio or Jupyter and import the final code as a new Galaxy tool and attach that to the workflow.

4. Both KNIME and Galaxy allow the workflow to be configured through configuration panels designed by the workflow creator. These panels form the user interface for that workflow. When these are well-annotated, it will provide a powerful clue to the novice about the details of the data processing and analysis.

5. KNIME nodes typically execute one or a few rows of the input table at a time, so that when the spectra and other project data are stored in a row-oriented fashion, it is not necessary to load the whole table into memory. This means that large projects can be handled by relatively modest hardware. In the case of Galaxy it depends on the implementation of the tools themselves how much computer resources are required.

6. Troubleshooting in KNIME involves checking the input, configuration and output of each node in the workflow and correct these if necessary. Data can be inspected directly by looking at the data table, which can be sorted in various ways. The code in R and Python nodes can be tweaked if necessary. Troubleshooting a Galaxy workflow is harder. While Galaxy is a lot more flexible with the types of data that can be passed from node to node, it relies on the tools themselves to present the data in a useful way. Although the tool configuration can be tweaked, code needs to be altered in external development tools.

7. KNIME allows the export and import of workflows, which can be easily archived or shared with other scientists. Alternatively, the virtual machine KIMBLE operates in can be shared with anyone with a VirtualBox installation. Similarly, Galaxy can be run from a Docker container which can be easily shared. Containers are typically considerably smaller than virtual machines, although running a (typically Linux-based) Galaxy container in Windows requires a Linux virtual machine to be running in the background.

Conclusions

The advance of workflow platforms provides highly needed reproducibility to the data processing and analysis pipeline. While workflows are already widely used in other omics fields, in metabolomics many scientists process their data with elaborate scripts in Jupyter or RStudio notebooks. It can be argued that these should not be considered workflow platforms, as code and data in these systems are separate entities so that the link between the raw source data and the final results can be easily lost. In contrast, KNIME and Galaxy are two workflow platforms that provide a close connection between data and algorithms. Both are the basis of versatile metabolomics workflows; in the case of KNIME these are KniMet and KIMBLE, while Workflow4Metabolomics and PheNoMeNal are built on top of Galaxy. Galaxy is a pure workflow platform and is already well established in the genomics field, while KNIME is also a development platform for R and Python scripts and allows for easy exploratory data analysis.

Conflicts of interest

The authors are the developers of the KIMBLE NMR metabolomics workflow.

References

1 Z. D. Stephens, S. Y. Lee, F. Faghri, R. H. Campbell, C. Zhai, M. J. Efron, R. Iyer, M. C. Schatz, S. Sinha and G. E. Robinson, *PLoS Biol.*, 2015, **13**, e1002195.
2 C. Chen, H. Huang and C. H. Wu, in *Protein Bioinformatics*, ed. C. H. Wu, C. N. Arighi and K. E. Ross, Springer New York, New York, NY, 2017, vol. 1558, pp. 3–39.
3 M. Wilhelm, J. Schlegl, H. Hahne, A. M. Gholami, M. Lieberenz, M. M. Savitski, E. Ziegler, L. Butzmann, S. Gessulat, H. Marx, T. Mathieson, S. Lemeer, K. Schnatbaum, U. Reimer, H. Wenschuh, M. Mollenhauer, J. Slotta-Huspenina, J.-H. Boese, M. Bantscheff, A. Gerstmair, F. Faerber and B. Kuster, *Nature*, 2014, **509**, 582–587.
4 M.-S. Kim, S. M. Pinto, D. Getnet, R. S. Nirujogi, S. S. Manda, R. Chaerkady, A. K. Madugundu, D. S. Kelkar, R. Isserlin, S. Jain, J. K. Thomas, B. Muthusamy, P. Leal-Rojas, P. Kumar, N. A. Sahasrabuddhe, L. Balakrishnan, J. Advani, B. George, S. Renuse, L. D. N. Selvan, A. H. Patil, V. Nanjappa, A. Radhakrishnan, S. Prasad, T. Subbannayya, R. Raju, M. Kumar, S. K. Sreenivasamurthy, A. Marimuthu, G. J. Sathe, S. Chavan, K. K. Datta, Y. Subbannayya, A. Sahu, S. D. Yelamanchi, S. Jayaram, P. Rajagopalan, J. Sharma, K. R. Murthy, N. Syed, R. Goel, A. A. Khan, S. Ahmad, G. Dey, K. Mudgal, A. Chatterjee, T.-C. Huang, J. Zhong, X. Wu, P. G. Shaw, D. Freed, M. S. Zahari, K. K. Mukherjee, S. Shankar, A. Mahadevan, H. Lam, C. J. Mitchell, S. K. Shankar, P. Satishchandra, J. T. Schroeder, R. Sirdeshmukh, A. Maitra, S. D. Leach, C. G. Drake, M. K. Halushka, T. S. K. Prasad, R. H. Hruban, C. L. Kerr, G. D. Bader, C. A. Iacobuzio-Donahue, H. Gowda and A. Pandey, *Nature*, 2014, **509**, 575–581.

5 G. J. Patti, O. Yanes and G. Siuzdak, *Nat. Rev. Mol. Cell Biol.*, 2012, **13**, 263–269.
6 M. L. Reaves and J. D. Rabinowitz, *Curr. Opin. Biotechnol.*, 2011, **22**, 17–25.
7 K. L. M. Coene, L. A. J. Kluijtmans, E. van der Heeft, U. F. H. Engelke, S. de Boer, B. Hoegen, H. J. T. Kwast, M. van de Vorst, M. C. D. G. Huigen, I. M. L. W. Keularts, M. F. Schreuder, C. D. M. van Karnebeek, S. B. Wortmann, M. C. de Vries, M. C. H. Janssen, C. Gilissen, J. Engel and R. A. Wevers, *J. Inherited Metab. Dis.*, 2018, **41**, 337–353.
8 P. Soininen, A. J. Kangas, P. Würtz, T. Suna and M. Ala-Korpela, *Circ.: Cardiovasc. Genet.*, 2015, **8**, 192–206.
9 J. Deelen, J. Kettunen, K. Fischer, A. van der Spek, S. Trompet, G. Kastenmüller, A. Boyd, J. Zierer, E. B. van den Akker, M. Ala-Korpela, N. Amin, A. Demirkan, M. Ghanbari, D. van Heemst, M. A. Ikram, J. B. van Klinken, S. P. Mooijaart, A. Peters, V. Salomaa, N. Sattar, T. D. Spector, H. Tiemeier, A. Verhoeven, M. Waldenberger, P. Würtz, G. Davey Smith, A. Metspalu, M. Perola, C. Menni, J. M. Geleijnse, F. Drenos, M. Beekman, J. W. Jukema, C. M. van Duijn and P. E. Slagboom, *Nat. Commun.*, 2019, **10**, 3346.
10 A.-H. Emwas, R. Roy, R. T. McKay, L. Tenori, E. Saccenti, G. A. N. Gowda, D. Raftery, F. Alahmari, L. Jaremko, M. Jaremko and D. S. Wishart, *Metabolites*, 2019, **9**, 123.
11 B. B. Misra and S. Mohapatra, *Electrophoresis*, 2019, **40**, 227–246.
12 *TopSpin*, Bruker BioSpin GmbH, Silberstreifen 4, 76287 Rheinstetten, Germany.
13 *Microsoft Excel*, Microsoft Corporation, Redmons WA, USA.
14 *SPSS Statistics*, IBM Corporation, Armonk NY, USA.
15 *SIMCA*, Umetrics/Sartorius-Stedim.
16 S. N. Goodman, D. Fanelli and J. P. A. Ioannidis, *Sci. Transl. Med.*, 2016, **8**, 341ps12.
17 C. G. Begley and L. M. Ellis, *Nature*, 2012, **483**, 531–533.
18 *Chenomx NMR Suite*, Chenomx Inc., 4232 - 10230 Jasper Ave, Edmonton, Alberta, Canada.
19 E. Melamud, L. Vastag and J. D. Rabinowitz, *Anal. Chem.*, 2010, **82**, 9818–9826.
20 M. M. Rinschen, J. Ivanisevic, M. Giera and G. Siuzdak, *Nat. Rev. Mol. Cell Biol.*, 2019, **20**, 353–367.
21 M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao and B. Mons, *Sci. Data*, 2016, **3**, 160018.
22 K. Haug, R. M. Salek, P. Conesa, J. Hastings, P. de Matos, M. Rijnbeek, T. Mahendraker, M. Williams, S. Neumann, P. Rocca-Serra, E. Maguire, A. González-Beltrán, S.-A. Sansone, J. L. Griffin and C. Steinbeck, *Nucleic Acids Res.*, 2013, **41**, D781–D786.
23 M. Sud, E. Fahy, D. Cotter, K. Azam, I. Vadivelu, C. Burant, A. Edison, O. Fiehn, R. Higashi, K. S. Nair, S. Sumner and S. Subramaniam, *Nucleic Acids Res.*, 2016, **44**, D463–D470.
24 A. Rowstron, D. Narayanan, A. Donnelly, G. O'Shea and A. Douglas, in *Proceedings of the 1st International Workshop on Hot Topics in Cloud Data Processing - HotCDP '12*, ACM Press, Bern, Switzerland, 2012, pp. 1–5.
25 W. W. Gibbs, *Nature*, 2017, **552**, 137–139.
26 J. M. Perkel, *Nature*, 2015, **518**, 125–126.
27 J. Chong, O. Soufan, C. Li, I. Caraus, S. Li, G. Bourque, D. S. Wishart and J. Xia, *Nucleic Acids Res.*, 2018, **46**, W486–W494.
28 J. Chong and J. Xia, *Bioinformatics*, 2018, **34**, 4313–4314.
29 P. Franceschi, R. Mylonas, N. Shahaf, M. Scholz, P. Arapitsas, D. Masuero, G. Weingart, S. Carlin, U. Vrhovsek, F. Mattivi and R. Wehrens, *Front. Bioeng. Biotechnol.*, 2014, **2**, 72.
30 S. Ravanbakhsh, P. Liu, T. C. Bjordahl, R. Mandal, J. R. Grant, M. Wilson, R. Eisner, I. Sinelnikov, X. Hu, C. Luchinat, R. Greiner and D. S. Wishart, *PLoS One*, 2015, **10**, e0124219.
31 A. Biswas, K. C. Mynampati, S. Umashankar, S. Reuben, G. Parab, R. Rao, V. S. Kannan and S. Swarup, *Bioinformatics*, 2010, **26**, 2639–2640.
32 J. Koster and S. Rahmann, *Bioinformatics*, 2012, **28**, 2520–2522.
33 E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. Ferreira da Silva, M. Livny and K. Wenger, *Future Gener. Comput. Syst.*, 2015, **46**, 17–35.
34 B. Balis, *Future Gener. Comput. Syst.*, 2016, **55**, 147–162.
35 P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo and C. Notredame, *Nat. Biotechnol.*, 2017, **35**, 316–319.
36 E. R. Gansner and S. C. North, *Software: Pract. Exper.*, 2000, **30**, 1203–1233.
37 M. Wang, J. J. Carver, V. V. Phelan, L. M. Sanchez, N. Garg, Y. Peng, D. D. Nguyen, J. Watrous, C. A. Kapono, T. Luzzatto-Knaan, C. Porto, A. Bouslimani, A. V. Melnik, M. J. Meehan, W.-T. Liu, M. Crüsemann, P. D. Boudreau, E. Esquenazi, M. Sandoval-Calderón, R. D. Kersten, L. A. Pace, R. A. Quinn, K. R. Duncan, C.-C. Hsu, D. J. Floros, R. G. Gavilan, K. Kleigrewe, T. Northen, R. J. Dutton, D. Parrot, E. E. Carlson, B. Aigle, C. F. Michelsen, L. Jelsbak, C. Sohlenkamp, P. Pevzner, A. Edlund, J. McLean, J. Piel, B. T. Murphy, L. Gerwick, C.-C. Liaw, Y.-L. Yang, H.-U. Humpf, M. Maansson, R. A. Keyzers, A. C. Sims, A. R. Johnson, A. M. Sidebottom, B. E. Sedio, A. Klitgaard, C. B. Larson, C. A. P. Boya, D. Torres-Mendoza, D. J. Gonzalez, D. B. Silva, L. M. Marques, D. P. Demarque, E. Pociute, E. C. O'Neill, E. Briand, E. J. N. Helfrich, E. A. Granatosky, E. Glukhov, F. Ryffel, H. Houson, H. Mohimani, J. J. Kharbush, Y. Zeng,

J. A. Vorholt, K. L. Kurita, P. Charusanti, K. L. McPhail, K. F. Nielsen, L. Vuong, M. Elfeki, M. F. Traxler, N. Engene, N. Koyama, O. B. Vining, R. Baric, R. R. Silva, S. J. Mascuch, S. Tomasi, S. Jenkins, V. Macherla, T. Hoffman, V. Agarwal, P. G. Williams, J. Dai, R. Neupane, J. Gurr, A. M. C. Rodríguez, A. Lamsa, C. Zhang, K. Dorrestein, B. M. Duggan, J. Almaliti, P.-M. Allard, P. Phapale, L.-F. Nothias, T. Alexandrov, M. Litaudon, J.-L. Wolfender, J. E. Kyle, T. O. Metz, T. Peryea, D.-T. Nguyen, D. VanLeer, P. Shinn, A. Jadhav, R. Müller, K. M. Waters, W. Shi, X. Liu, L. Zhang, R. Knight, P. R. Jensen, B. Ø. Palsson, K. Pogliano, R. G. Linington, M. Gutiérrez, N. P. Lopes, W. H. Gerwick, B. S. Moore, P. C. Dorrestein and N. Bandeira, *Nat. Biotechnol.*, 2016, **34**, 828–837.

38 L. F. Nothias, D. Petras, R. Schmid, K. Dührkop, J. Rainer, A. Sarvepalli, I. Protsyuk, M. Ernst, H. Tsugawa, M. Fleischauer, F. Aicheler, A. Aksenov, O. Alka, P.-M. Allard, A. Barsch, X. Cachet, M. Caraballo, R. R. Da Silva, T. Dang, N. Garg, J. M. Gauglitz, A. Gurevich, G. Isaac, A. K. Jarmusch, Z. Kameník, K. B. Kang, N. Kessler, I. Koester, A. Korf, A. L. Gouellec, M. Ludwig, M. H. Christian, L.-I. McCall, J. McSayles, S. W. Meyer, H. Mohimani, M. Morsy, O. Moyne, S. Neumann, H. Neuweger, N. H. Nguyen, M. Nothias-Esposito, J. Paolini, V. V. Phelan, T. Pluskal, R. A. Quinn, S. Rogers, B. Shrestha, A. Tripathi, J. J. J. van der Hooft, F. Vargas, K. C. Weldon, M. Witting, H. Yang, Z. Zhang, F. Zubeil, O. Kohlbacher, S. Böcker, T. Alexandrov, N. Bandeira, M. Wang and P. C. Dorrestein, *bioRxiv*, 2019, 812404.

39 K. Wolstencroft, R. Haines, D. Fellows, A. Williams, D. Withers, S. Owen, S. Soiland-Reyes, I. Dunlop, A. Nenadic, P. Fisher, J. Bhagat, K. Belhajjame, F. Bacall, A. Hardisty, A. Nieva de la Hidalga, M. P. Balcazar Vargas, S. Sufi and C. Goble, *Nucleic Acids Res.*, 2013, **41**, W557–W561.

40 J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik and B. Zupan, *J. Mach. Learn. Res.*, 2013, **14**, 2349–2353.

41 B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao and Y. Zhao, *Concurr. Comput.: Pract. Exper.*, 2006, **18**, 1039–1065.

42 I. Altintas, J. Wang, D. Crawl and W. Li, in *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, ACM Press, Berlin, Germany, 2012, pp. 73–78.

43 *RapidMiner Studio*, RapidMiner Inc., Boston MA, USA.

44 *Pipeline Pilot*, BIOVIA, San Diego CA, United States.

45 *Alteryx Designer*, Alteryx Inc., Irvine CA, USA.

46 E. Afgan, D. Baker, B. Batut, M. van den Beek, D. Bouvier, M. Čech, J. Chilton, D. Clements, N. Coraor, B. A. Grüning, A. Guerler, J. Hillman-Jackson, S. Hiltemann, V. Jalili, H. Rasche, N. Soranzo, J. Goecks, J. Taylor, A. Nekrutenko and D. Blankenberg, *Nucleic Acids Res.*, 2018, **46**, W537–W544.

47 M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel and B. Wiswedel, in *Data Analysis, Machine Learning and Applications*, Springer, 2008, pp. 319–326.

48 A. Fillbrunn, C. Dietz, J. Pfeuffer, R. Rahn, G. A. Landrum and M. R. Berthold, *J. Biotechnol.*, 2017, **261**, 149–156.

49 B. A. Grüning, E. Rasche, B. Rebolledo-Jaramillo, C. Eberhard, T. Houwaart, J. Chilton, N. Coraor, R. Backofen, J. Taylor and A. Nekrutenko, *PLoS Comput. Biol.*, 2017, **13**, e1005425.

50 G. Landrum, *KNIME and Jupyter*, https://www.knime.com/blog/knime-and-jupyter, (accessed 30 August 2019).

51 Y. Guitton, M. Tremblay-Franco, G. Le Corguillé, J.-F. Martin, M. Pétéra, P. Roger-Mele, A. Delabrière, S. Goulitquer, M. Monsoor, C. Duperier, C. Canlet, R. Servien, P. Tardivel, C. Caron, F. Giacomoni and E. A. Thévenot, *Int. J. Biochem. Cell Biol.*, 2017, **93**, 89–101.

52 K. Peters, J. Bradbury, S. Bergmann, M. Capuccini, M. Cascante, P. de Atauri, T. M. D. Ebbels, C. Foguet, R. Glen, A. Gonzalez-Beltran, U. L. Günther, E. Handakas, T. Hankemeier, K. Haug, S. Herman, P. Holub, M. Izzo, D. Jacob, D. Johnson, F. Jourdan, N. Kale, I. Karaman, B. Khalili, P. Emami Khonsari, K. Kultima, S. Lampa, A. Larsson, C. Ludwig, P. Moreno, S. Neumann, J. A. Novella, C. O'Donovan, J. T. M. Pearce, A. Peluso, M. E. Piras, L. Pireddu, M. A. C. Reed, P. Rocca-Serra, P. Roger, A. Rosato, R. Rueedi, C. Ruttkies, N. Sadawi, R. M. Salek, S.-A. Sansone, V. Selivanov, O. Spjuth, D. Schober, E. A. Thévenot, M. Tomasoni, M. van Rijswijk, M. van Vliet, M. R. Viant, R. J. M. Weber, G. Zanetti and C. Steinbeck, *GigaScience*, 2019, **8**, giy149.

53 P. Emami Khoonsari, P. Moreno, S. Bergmann, J. Burman, M. Capuccini, M. Carone, M. Cascante, P. de Atauri, C. Foguet, A. N. Gonzalez-Beltran, T. Hankemeier, K. Haug, S. He, S. Herman, D. Johnson, N. Kale, A. Larsson, S. Neumann, K. Peters, L. Pireddu, P. Rocca-Serra, P. Roger, R. Rueedi, C. Ruttkies, N. Sadawi, R. M. Salek, S.-A. Sansone, D. Schober, V. Selivanov, E. A. Thévenot, M. van Vliet, G. Zanetti, C. Steinbeck, K. Kultima and O. Spjuth, *Bioinformatics*, 2019, **35**, 3752–3760.

54 Y. Perez-Riverol and P. Moreno, *Proteomics*, 2019, 1900147.

55 R. L. Davidson, R. J. M. Weber, H. Liu, A. Sharma-Oates and M. R. Viant, *GigaScience*, 2016, **5**, 10.

56 A. Verhoeven, M. Giera and O. A. Mayboroda, *Anal. Chim. Acta*, 2018, **1044**, 66–76.

57 S. Liggi, C. Hinz, Z. Hall, M. L. Santoru, S. Poddighe, J. Fjeldsted, L. Atzori and J. L. Griffin, *Metabolomics*, 2018, **14**, 52.

58 J. J. Helmus and C. P. Jaroniec, *J. Biomol. NMR*, 2013, **55**, 355–367.

59 J. Lewis, C. E. Breeze, J. Charlesworth, O. J. Maclaren and J. Cooper, *BMC Syst. Biol.*, 2016, **10**, 52.

60 H. L. Röst, T. Sachsenberg, S. Aiche, C. Bielow, H. Weisser, F. Aicheler, S. Andreotti, H.-C. Ehrlich, P. Gutenbrunner, E. Kenar, X. Liang, S. Nahnsen, L. Nilse, J. Pfeuffer, G. Rosenberger, M. Rurik, U. Schmitt, J. Veit, M. Walzer, D. Wojnar, W. E. Wolski, O. Schilling, J. S. Choudhary, L. Malmström, R. Aebersold, K. Reinert and O. Kohlbacher, *Nat. Methods*, 2016, **13**, 741–748.

