From the desktop to the grid: conversion of KNIME Workflows to gUSE

Luis de la Garza
Applied Bioinformatics Group
University of Tübingen, Germany
delagarza@informatik.uni-tuebingen.de

Jens Krüger
Applied Bioinformatics Group
University of Tübingen, Germany

Charlotta Schärfe
Applied Bioinformatics Group
University of Tübingen, Germany

Marc Röttig
Applied Bioinformatics Group
University of Tübingen, Germany

Stephan Aiche
Department of Mathematics and Computer Science
Freie Universität Berlin, Germany
International Max Planck Research School
for Computational Biology and Scientific Computing
Berlin, Germany

Knut Reinert
Algorithms in Bioinformatics
Freie Universität Berlin, Germany

Oliver Kohlbacher
Applied Bioinformatics Group
University of Tübingen, Germany
oliver.kohlbacher@uni-tuebingen.de

***Abstract*—The Konstanz Information Miner is a user-friendly graphical workflow designer with a broad user base in industry and academia. Its broad range of embedded tools and its powerful data mining and visualization tools render it ideal for scientific workflows. It is thus used more and more in a broad range of applications. However, the free version typically runs on a desktop computer, restricting users if they want to tap into computing power. The grid and cloud User Support Environment is a free and open source project created for parallelized and distributed systems, but the creation of workflows with the included components has a steeper learning curve.**

In this work we suggest an easy to implement solution combining the ease-of-use of the Konstanz Information Miner with the computational power of distributed computing infrastructures. We present a solution permitting the conversion of workflows between the two platforms. This enables a convenient development, debugging, and maintenance of scientific workflows on the desktop. These workflows can then be deployed on a cloud or grid, thus permitting large-scale computation.

To achieve our goals, we relied on a Common Tool Description XML file format which describes the execution of arbitrary programs in a structured and easily readable and parseable way. In order to integrate external programs into we employed the Generic KNIME Nodes extension.

I. Introduction

Workflow technology with platforms such as Pipeline Pilot [1], KNIME [2], Taverna [3], [4], [5] and Galaxy [6], [7], [8] have now become a crucial part in supporting scientists in their daily work. By helping to create and automate virtual processes such as molecular docking or molecular dynamics simulations, as well as simplifying data analysis and data mining, scientists are allowed to focus on their primary goals [9]. Furthermore the quality of simulation results is improved, as following established protocols increases reproducibility in the sense of good lab practice.

The most obvious and direct advantage of the application of workflows in the scientific environment is the capability of saving the general sequence of events in order to conveniently optimize the settings for a simulation, such as including the sweep through single parameter settings. Scientists also benefit from other non-obvious advantages of using workflows including, but not limited to: ability to analyze the results, including statistical analysis and data visualization, data mining on experimentally (wet or dry lab) obtained datasets and report creation using previously obtained data without requiring further user input.

Those tasks can also be fulfilled using simple scripts or separate program suites for the individual steps. Workflow technology however allows combining all steps together by providing interfaces to external tools while not requiring any knowledge of programming or scripting languages. Additionally, the workflows established within one project may be easily applied to other projects as well, which then facilitates consistency in analysis and reporting throughout several projects, thus reducing the risk of human error and allowing reproducing previous results. Furthermore, through the ability to share workflows with collaborators or the scientific community a team-based analysis of experimental results can take place.

Nowadays a plethora of different workflow systems exists that was initially targeted at different use cases such as desktop-based data mining or automation of computations on a grid.

With the exponential increase of computational power [10] available to scientists, as well as the improvements in network technology, the boundaries between local applications and processes executed on distributed systems became blurred. This has as a result that there does not yet exist a one-fits-all solution that is being able to satisfy the scientific user's needs for a combination of local and distributed workflow execution. In addition to that, most users of workflow technology in the scientific environment have created a library of own workflows with their workflow suite of choice over the past years. These may now be outdated or not suited for the computation resources required for today's tasks, thus requiring the switch to another workflow environment and the need of re-implementing the existing workflows in the workflow language used by the new environment. For example, the Konstanz Information Miner (KNIME) [2] was mainly created for applications on a local machine and its free version does not provide access to compute clusters out of the box, but KNIME has, due to its ease of use and extensibility, found wide acceptance in the scientific community resulting in a huge library of existing KNIME workflows for various tasks. The grid and cloud User Support Environment (gUSE) [11] on the other hand, was specifically created to use distributed computing infrastructures (DCI), but the creation of workflows requires more user input and therefore is not as straightforward as local systems such as KNIME. A KNIME user now may want to not only use the KNIME desktop version for data analysis and pilot runs for evaluating simulation parameters and post-simulation analysis, but also the open source gUSE environment for moving the actual simulations to a cluster. The workflows for the simulation pilot run and the actual simulation are identical since the first is used to find the best settings and the latter then applies those settings. When using two different software suites such as KNIME and gUSE for the pilot run and the actual full-scale simulation, it is currently required to implement the workflow twice (one for each software). The same applies when switching the workflow software. This re-implementation of existing workflows is a tedious task that would not be needed if it were possible to convert workflows written with one workflow language in a way that it could then be read by another workflow environment - thus enabling workflow interoperability.

II. Related Work

The question whether a certain computational task is executable on different platforms is as old as computers themselves. Regarding modern workflow languages, a couple of specific challenges come into focus, discussed in detail in the following chapters. Since there is a multitude of workflow languages, the focus shifts for different use cases and other user communities. The most prominent approach to deal with the general problem of workflow interoperability is SHIWA and its follow up project ER-FLOW [12], [13]. A double strategy was followed, namely coarse and fine grained interoperability. The first one considers a workflow language as a black box, enabling the execution of sub-workflows within WS-PGRADE, which acts as a host system. The second approach puts emphasis on the actual transformation of selected workflow languages such as ASKALON, Pegasus, P-GRADE, MOTEUR and Triana into each other [14]. ER-FLOW continues these ideas, adding the aspects of detailed evaluation of user community needs and the specific handling of scientific applications on remote DCIs, being called by the workflows.

III. Workflows

The concept of recipes or protocols is familiar to scientists from all academic fields. The expression "workflow" follows this concept of a collection of consecutive computational steps. This may involve preparation steps for importing data, converting it and to carrying out whatever preparations are required. After these steps the actual simulation or computational step is usually carried out. There is a multitude of possible application domains, like quantum calculations, molecular dynamics, docking or data mining to name only a few. The last section of a typical workflow deals with the data analysis and visualization, often summarized in the form of a report.

An important aspect for workflow interoperability is the representation as a graph. The individual tasks represent the nodes; their edges correspond to the data flow or execution dependencies between these nodes. Hence, when a workflow shall be converted from e.g. KNIME to gUSE it has to be taken care that the graph representation is similar. Is the workflow represented as a strict directed graph or does it correspond to a multigraph? Are parameter sweeps executable via loops or through the enumeration of predefined lists? Does the workflow have multiple start or end points corresponding to a quiver? This small selection of questions illustrates the logical constraints faced when dealing with workflow conversion from one language into another. Furthermore the data handling and its flow along the graph is of relevance. Is the data directly incorporated into the nodes, e.g. as tables or does it reside elsewhere independently of the execution status of the specific node? Are there specific formats or conventions regarding the dependency to the workflow language? How is the data annotated? Great care has to be taken when facing the conversion of data from different workflow languages.

In the following chapters specific details of KNIME and gUSE are described.

IV. KNIME

KNIME is one of the most commonly used workflow management systems in the field of e-Science systems, especially pharmaceutical research, but also financial data analysis and business intelligence [15]. The KNIME pipelining platform is an open-source program implemented as a plug-in for Eclipse [2], written in Java, and offered to the scientific community as a desktop version free of charge. Although

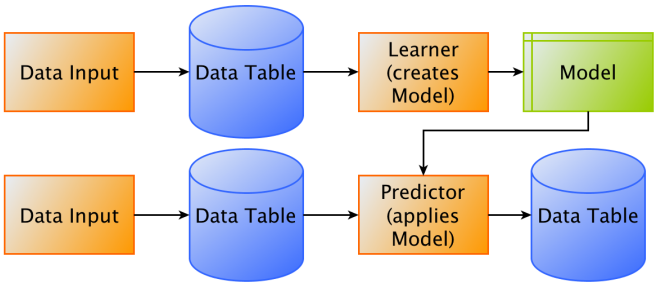

Fig. 1. An illustration of the KNIME workflow concept. Nodes represent single processing unites and connecting edges between these nodes transport data or models from one processing unit to the next. In the end a final data table is created that can be saved to a file. Figure modified from [19]

there are extensions allowing the execution of single nodes in the cloud or on a grid [16], these are only restricted to a professional release and are thus not part of the free workflow management system. Furthermore, KNIME is highly popular due to its easy of use and extensibility.

The KNIME platform implements a modular approach to workflow management and execution in which single nodes represent single processing units such as data manipulation, as depicted in Figure 1. These nodes are connected via edges that pipe either data or computational models from one node into the next node. Data is internally stored in special java classes called *DataTable*, which store the data and additional meta information about the different data columns [17], [18]. The nodes and edges together form a directed acyclic graph, which is called ”workflow” and converts initial input files into output data tables that can be further exported as new files [18].

The implementation as an Eclipse plug-in with its free API [20] facilitates easy extensibility of the workflow system and simple integration of novel nodes thus resulting in a vast library of nodes created by the scientific community and also commercial software providers.

V. GUSE

The grid and cloud User Support Environment (gUSE) [11] is a highly popular technology for scientific portals enabling access to distributed computing infrastructures (DCIs). It has been developed at the Laboratory of Parallel and Distributed Systems in Budapest over the past years. gUSE represents the middle tier of a multi-layer portal solution. Different tasks can be handled by a set of high level web services (see Figure 2). The Application Repository holds the executable for all programs that may be linked to a node within a workflow. The File Storage deals with the data handling, while the Information System takes care of e.g. user information and job status. The Workflow Interpreter is responsible for the workflows and their execution, which are stored in the Workflow Storage. The Submitter represents the connection

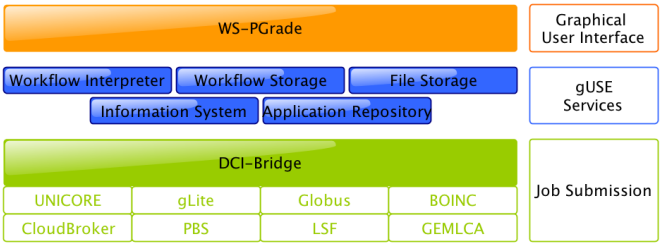

Fig. 2. The layered structure of gUSE/WS-PGRADE is shown. Figure modified from [21]

between gUSE services and middlewares, enabling access to the computational resources of grid or cloud. On the top layer resides WS-PGRADE, the graphical user interface. All functionality of the underlying services is exposed to the end-user by portlets residing in a Liferay portlet container being part of WS-PGRADE.

gUSE workflows may be created and maintained via standard web browsers accessing corresponding portlets and underlying services. Initially the workflow graph has to be created through a Java applet. The nodes have to be defined while each node may have multiple input and output ports. These work as anchor points for the vertices connecting them. The selection of applications is done through the Concrete portlet also enabling the selection of different DCIs with different middlewares within the same workflow. Application specific parameters can be set, as well as resource requirements such as memory or runtime settings. Beside submission and monitoring features, a multitude of import and export features are available to the user.

The whole set of services offers convenient access to the vast computational resources of modern grids and clouds. gUSE is available free of charge for academic purposes.

VI. GENERIC KNIME NODES

As previously discussed, KNIME offers a wide array of prebuilt nodes for the execution of a multitude of different tasks. It is also possible to obtain external nodes provided by community developers such as the ones developed by Schrödinger [22], ChemAxon [23], etc. Furthermore, it is possible to develop KNIME nodes, being a simple task of implementing a few KNIME specific classes in the Java programming language. However, we still felt that, although KNIME is powerful for most computations and it enables users to easily extend its capabilities, sometimes it is needed to integrate external binaries into KNIME in the form of a node in a simpler way.

We used a KNIME extension called Generic KNIME Nodes [24], [25], which allows the integration of arbitrary programs into KNIME. This integration is fully compatible with KNIME and other KNIME nodes and each integrated program behaves as a KNIME node. Since KNIME relies on

the use of data tables rather than on files, GKN also includes utility nodes such as File to Table, Table to File, Input File and Output File to ease the interaction of a GKN-generated node with other nodes.

In order for GKN to properly execute external binaries, we also relied on an XML-based file format that describes tools, nodes of the workflow graph, called Common Tool Description (CTD) [26]. CTD files are XML documents that contain information about the parameters, flags, inputs and outputs of a given binary. This information is presented in a structured and human readable way, thus facilitating manual generation for arbitrary binaries. Since CTDs are also properly formed XML documents, parsing of these is a trivial matter.

The generation of CTDs can be either manual or by CTD capable programs. Software tool suites such as SeqAn [27], OpenMS [28] and CADDSuite [29] can not only generate CTDs for each of its tools, but can also parse input CTDs and execute its tools accordingly.

VII. Conversion from KNIME to gUSE

A. Overview

The motivation for this conversion lies in the fact that most scientific computations can be memory and processor intensive. The requirements to run such computations in an acceptable time frame are hardly to be met by a simple desktop or laptop computer. Grids and clouds are packed with resources ready to be tapped, but as earlier discussed; creating workflows on such systems can be a tedious task that the most enthusiast scientists might not be ready to go through. Based on the popularity of KNIME's ease of use and its wide acceptance in the scientific community, we felt that there was a gap to be filled by bridging a great workflow editor such as KNIME with a great grid and cloud manager such as gUSE.

Our vision is to have users creating and executing workflows on KNIME in their desktop computers using a reduced or a test dataset and when an acceptable stable version of a workflow is ready, it can be exported into a gUSE managed grid or cloud. Following this, the user would have to configure the exported workflow to include a larger or a production-ready dataset on which to perform a computation.

One of the great features of the Eclipse Platform is its extendibility through the development of so called plug-ins [30]. Given that KNIME has been built on top of the Eclipse Platform, it is fairly simple to develop KNIME extensions, which in turn are Eclipse Platform plug-ins. KNIME also exposes an API that gives full access to all of the elements involved in a workflow [20], both visually and logically. We have developed a simple conversion KNIME extension that can export a KNIME Workflow to gUSE format.

The critical challenge for workflow conversion arises when it comes to the translation of the configuration that assists in the execution of the workflow. It is clear that significant effort has to be invested to resolve any potential disparity between the architectures on the computer in which the KNIME workflow was created and the grid or cloud. In other words, a generic solution cannot simply rely on both the desktop machine in which KNIME is being executed and each node in the infrastructure administered by gUSE having the same architecture and therefore, the same binaries. For this reason, a conversion table relating the binaries needed for each step on the desktop to the ones required on the grid or cloud is needed. Since gUSE supports several middlewares (e.g. UNICORE, LSF, BOINC, etc.), the usage of a different format to represent the required information to execute a needed binary has to be accounted for in the workflow conversion process.

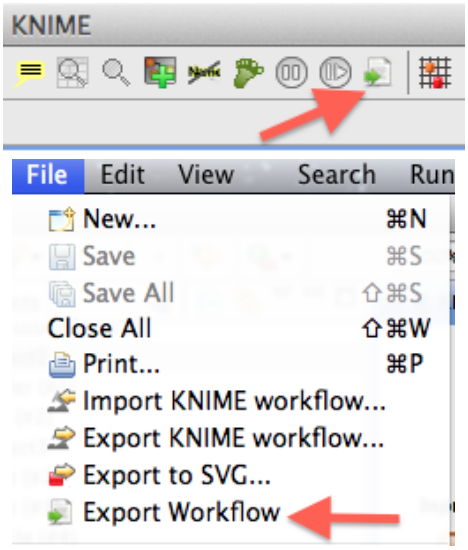

Fig. 3. In order to start the conversion of a workflow, we've integrated visual elements in the KNIME platform

B. Conversion of complete Workflows

The KNIME extension that we have developed to convert KNIME workflows to gUSE format is fully integrated in KNIME. When a user is satisfied with a certain workflow, all is needed is to request a conversion by simply clicking a button in a toolbar or a menu element (see Figure 3). What follows is a standard dialog window (see Figure 4) in which the user can select the desired destination to export the workflow. Once the user has selected an export destination, an archive that can be uploaded and imported into a gUSE portal will be generated.

The conversion process starts by using KNIME's API to access each node and its connections to convert them into a workflow in an intermediate, internal format. Afterwards, this internal format workflow is converted into a gUSE workflow. This seemingly impractical design choice was taken in order to follow the Separation of Concerns principle [31]. Since the

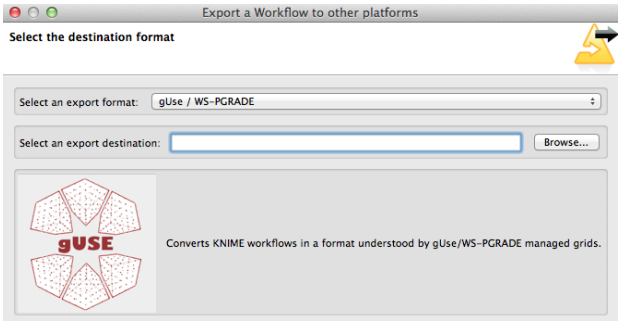

Fig. 4. An archive in the gUSE format will be generated, which can be imported into gUSE

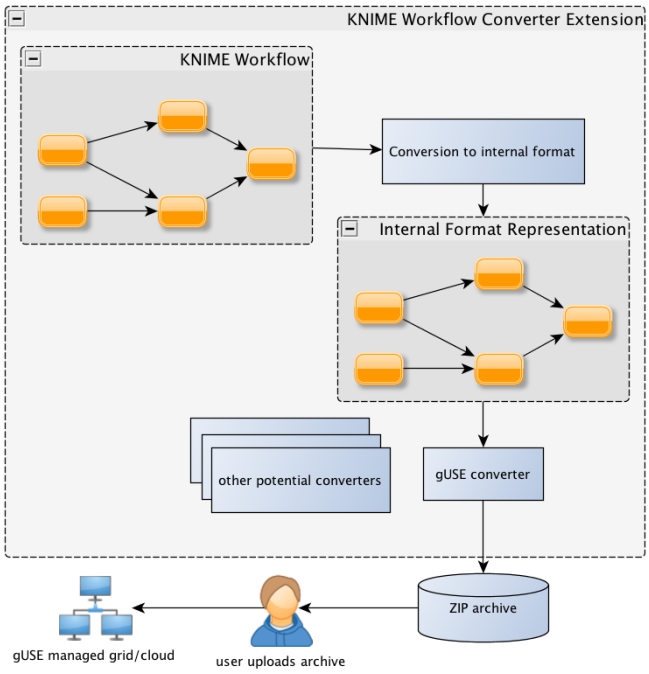

Fig. 5. Our KNIME extension has been designed taken into account extendability and modularity

release schedule of KNIME is something not under of our control, it is a good idea to minimize the exposure of the components of our conversion process from changes in KNIME's API or workflow format by first using an intermediate format. This intermediate format is something internal to the conversion process whose changes are mandated exclusively by us. Another advantage of this design is that, in the event of extending our KNIME extension by adding other export formats, it would only be needed to perform the conversion from this internal format without explicitly converting the KNIME workflow, thus, decreasing development time and the amount of code needed to perform the required task. This is broadly depicted on Figure 5.

C. Example Application: Docking Workflows

As users and developers of the Molecular Simulation Grid portal (MoSGrid) [32] we have learned not only to

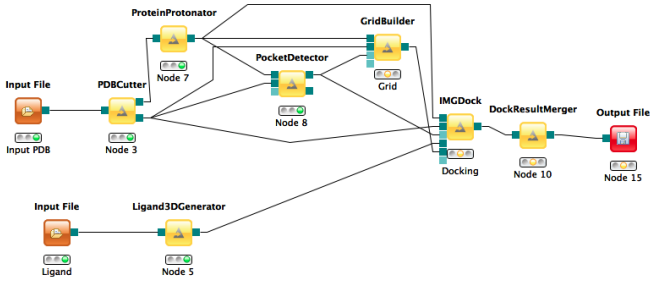

Fig. 6. Using GKN it is possible to perform docking in KNIME with the CADDSuite

benefit from the strengths of gUSE, but also to overcome its drawbacks. Part of our effort consisted of the creation of a docking workflow using the provided tools by gUSE. As mentioned in a previous section, getting a complex workflow right on gUSE can turn into a quite intimidating task for the inexperienced user. During this time, we felt that the creation of such a complex workflow could and should be simpler.

We perform docking using our own software, the Computer Aided Drug Design Suite (CADDSuite) [29], which we also integrated in KNIME using Generic KNIME Nodes, as depicted in Figure 6.

Putting this workflow together on KNIME took us less than an hour. A similar version of this workflow on gUSE took us significantly more than that. We were able to export the workflow to gUSE with minimal configuration, that is, we just needed to provide adequate input data files. Since docking is a processor intensive task, we used different data sets in our desktop computers and on the MoSGrid portal.

In order to use input files in KNIME with GKN, it is required to use the *Input File* node. Similarly, for output files, the *Output File* node must be used. However, in gUSE input and output files are directly associated to a job's input and output ports, respectively. This is the reason why during the conversion of a workflow from KNIME to gUSE any input or output file nodes will disappear and take the form of input or output ports in gUSE. (see Figure 7)

VIII. Future Work

A major work in progress is how to properly export workflows that benefit from parameter sweep. This is critical for the performance of exported workflows, since gUSE offers parallelization via parameter sweep.

KNIME offers several data mining, statistics and reporting nodes that could be easily integrated with our docking workflow. For instance, it would be desirable to generate a concise PDF report containing the top ranked ligands. Unfortunately, the conversion of such nodes is still not possible. However, KNIME offers a headless execution of workflows (i.e., command line), thus giving us the chance to

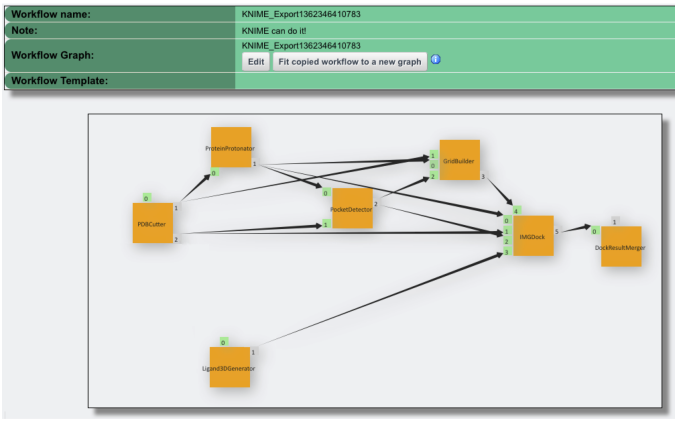

Fig. 7. The exported workflow in MoSGrid after the user has uploaded to the portal

work around this current limitation.

MoSGrid relies on UNICORE to access binaries and data. A workflow using UNICORE resources has a different representation in gUSE than a workflow using an LSF scheduler. Since we want to reach as many users as possible, it is desired that our KNIME extension can properly handle as many constellation of gUSE components as possible.

IX. Conclusion

Any robust scientific experiment must be repeatable. Workflow technologies provide their users with repeatability on the tasks that comprise a workflow. Furthermore, these technologies offer adopters with the possibility of saving temporary and final results for further analysis as well as the chance of rerunning a subset of tasks contained in a workflow. If a configuration error is detected in one of the tasks that make up a workflow, this very ability of storing intermediate results allow users to make changes to the configuration and later on resume the execution of the workflow without having to execute tasks not influenced by these changes.

Making grids and clouds accessible to users has the benefit of speeding up experiments, production of scientific texts, ensure an optimal use of resources and minimize idle computing time. As we have argued, one of the main obstacles in accessing grids and clouds is the steep learning curve to generate usable workflows. However, gUSE is accessible to users and excels in executing workflows in an efficient way.

It is far more easier to train users to use KNIME in order to generate workflows and test experiments than to teach them how to generate scripts for a certain resource manager or middleware. Using KNIME, users can rapidly generate a workflow by using an intuitive and robust user interface. The obvious limitation is that KNIME will have as much computing power as the desktop computer on which it runs and this might not be adequate for applications such as docking.

We have these two complementary forces that we feel our KNIME extension smoothly combines. On one side, we have gUSE enabling users to harness the power supplied by a grid or a cloud. On the other side, we have KNIME allowing users to create workflows in a friendly manner. Joining these two is of critical importance for the advancement of scientific fields in which an experiment can be broken up in smaller tasks to form a workflow.

Acknowledgment

The authors would like to thank the BMBF (German Federal Ministry of Education and Research) for the opportunity to do research in the MoSGrid project (reference 01IG09006). The research leading to these results has also partially been supported by the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 283481 (SCI-BUS).

Stephan Aiche gratefully acknowledges funding by the European Commissions's seventh Framework Program (GA263215).

References

[1] accelrys. Pipeline Pilot. [Online]. Available: http://accelrys.com/products/pipeline-pilot/

[2] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel, *KNIME: The Konstanz information miner*. Springer, 2008.

[3] P. Missier, S. Soiland-Reyes, S. Owen, W. Tan, A. Nenadic, I. Dunlop, A. Williams, T. Oinn, and C. Goble, "Taverna, reloaded," in *Scientific and Statistical Database Management*. Springer, 2010, pp. 471–481.

[4] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn, "Taverna: a tool for building and running workflows of services," *Nucleic acids research*, vol. 34, no. suppl 2, pp. W729–W732, 2006.

[5] T. Oinn, M. Greenwood, M. Addis, M. N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. Marvin *et al.*, "Taverna: lessons in creating a workflow environment for the life sciences," *Concurrency and Computation: Practice and Experience*, vol. 18, no. 10, pp. 1067–1100, 2006.

[6] J. Goecks, A. Nekrutenko, J. Taylor, T. G. Team *et al.*, "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences," *Genome Biol*, vol. 11, no. 8, p. R86, 2010.

[7] D. Blankenberg, G. V. Kuster, N. Coraor, G. Ananda, R. Lazarus, M. Mangan, A. Nekrutenko, and J. Taylor, "Galaxy: A web-based genome analysis tool for experimentalists," *Current protocols in molecular biology*, pp. 19–10, 2010.

[8] B. Giardine, C. Riemer, R. C. Hardison, R. Burhans, L. Elnitski, P. Shah, Y. Zhang, D. Blankenberg, I. Albert, J. Taylor *et al.*, "Galaxy: a platform for interactive large-scale genome analysis," *Genome research*, vol. 15, no. 10, pp. 1451–1455, 2005.

[9] K. Görlach, M. Sonntag, D. Karastoyanova, F. Leymann, and M. Reiter, "Conventional workflow technology for scientific simulation," in *Guide to e-Science*. Springer, 2011, pp. 323–352.

[10] G. E. Moore *et al.*, "Cramming more components onto integrated circuits," 1965.

[11] MTA-SZTAKI LPDS. grid and cloud User Support Environment. [Online]. Available: http://guse.hu/

[12] Sharing Interoperable Workflows for large-scale scientific Simulations on available DCIs. [Online]. Available: http://www.shiwa-workflow.eu/

[13] Building an European Research Community through Interoperable Workflows and Data. [Online]. Available: http://www.erflow.eu/

[14] M. Kozlovszky, K. Karoczkai, I. Marton, A. Balasko, A. Marosi, and P. Kacsuk, "Enabling generic distributed computing infrastructure compatibility for workflow management systems," *Computer Science*, vol. 13, no. 3, pp. 61–78, 2012.

[15] K. Achilleos, C. Kannas, C. Nicolaou, C. Pattichis, and V. Promponas, "Open source workflow systems in life sciences informatics," in *Bioinformatics & Bioengineering (BIBE), 2012 IEEE 12th International Conference on*. IEEE, 2012, pp. 552–558.

[16] CloudBroker: High Performance Computing Software as a Service - Integration in KNIME. [Online]. Available: http://www.knime.org/files/10_CloudBroker.pdf

[17] M. R. Berthold, N. Cebron, F. Dill, G. D. Fatta, T. R. Gabriel, F. Georg, T. Meinl, P. Ohl, C. Sieb, and B. Wiswedel, "Knime: The konstanz information miner," in *Proceedings of the Workshop on Multi-Agent Systems and Simulation (MAS&S), 4th Annual Industrial Simulation Conference (ISC)*, 2006, pp. 58–61.

[18] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, and B. Wiswedel, "Knime - the konstanz information miner: version 2.0 and beyond," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 26–31, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656280

[19] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel, "Knime: The konstanz information miner," in *In Data Analysis, Machine Learning and Applications - Proceedings of the 31st Annual Conference of the Gesellschaft für Klassifikation e.V., Studies in Classification, Data Analysis, and Knowledge Organization*. Berlin, Germany: Springer, 2007, pp. 319–326.

[20] KNIME Application Programming Interface. [Online]. Available: http://tech.knime.org/docs/api/

[21] MTA-SZTAKI LPDS. grid and cloud User Support Environment Architecture. [Online]. Available: http://www.guse.hu/?m=architecture&s=0

[22] Schrödinger KNIME Extensions. [Online]. Available: http://www.schrodinger.com/productpage/14/8/

[23] ChemAxon's JChem Nodes on the KNIME workbench. [Online]. Available: http://www.chemaxon.com/library/chemaxons-jchem-nodes-on-the-knime-workbench/

[24] M. Röttig, "Combining sequence and structural information into predictors of enzymatic activity," Ph.D. dissertation, University of Tübingen, Nov. 2012.

[25] M. Röttig, S. Aiche, L. de la Garza, and B. Kahlert. Generic KNIME Nodes. [Online]. Available: https://github.com/genericworkflownodes/GenericKnimeNodes

[26] OpenMS Team. Common Tool Description. [Online]. Available: http://open-ms.sourceforge.net/workflow-integration/topp-and-common-tool-description/

[27] SeqAn Team. SeqAn. [Online]. Available: http://www.seqan.de/

[28] OpenMS Team. OpenMS. [Online]. Available: http://open-ms.sourceforge.net/

[29] BALL Team. Computer Aided Drug Design Suite - CADDSuite. [Online]. Available: http://www.ball-project.org/caddsuite

[30] The Eclipse Foundation. Plug-In Development Environment. [Online]. Available: http://www.eclipse.org/pde/

[31] E. W. Dijkstra, "On the role of scientific thought," in *Selected Writings on Computing: A Personal Perspective*. Springer, 1982, pp. 60–66.

[32] S. Gesing, R. Grunzke, J. Krüger, G. Birkenheuer, M. Wewior, P. Schäfer, B. Schuller, J. Schuster, S. Herres-Pawlis, S. Breuers, A. Balaskó, M. Kozlovszky, A. S. Fabri, L. Packschies, P. Kacsuk, D. Blunk, T. Steinke, A. Brinkmann, G. Fels, R. Müller-Pfefferkorn, R. Jäkel, and O. Kohlbacher, "A Single Sign-On Infrastructure for Science Gateways on a Use Case for Structural Bioinformatics," *Journal of Grid Computing*, vol. 10, no. 4, pp. 769–790, Nov. 2012. [Online]. Available: http://www.springerlink.com/index/10.1007/s10723-012-9247-y