A Method to Build and Analyze Scientific Workflows from Provenance through Process Mining

Reng Zeng, Xudong He
Computing and Information Sciences
Florida International University
Miami, Florida 33199, USA
Email: {rzeng001, hex}@cis.fiu.edu

Jiafei Li
Computer Science and Technology
JiLin University
Changchun, 130012, China
Email: jiafei@jlu.edu.cn

Zheng Liu, W.M.P. van der Aalst
Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
Email: {z.liu3, w.m.p.v.d.aalst}@tue.nl

***Abstract*—Scientific workflows have recently emerged as a new paradigm for representing and managing complex distributed scientific computations and are used to accelerate the pace of scientific discovery. In many disciplines, individual workflows are large due to the large quantities of data used. As scientific workflows scale quickly, they become very hard to build and maintain. Recent efforts from scientific workflow community aiming at large-scale capturing of provenance present a new opportunity for building scientific workflows using provenance. Process mining focusses on extracting information about processes by examining event logs, and has been successfully applied to business workflow management. This paper presents a method using process mining based on provenance to build and analyze scientific workflows, which provides a new direction in using captured provenance.**

I. Introduction

Scientific computing has entered a new era of large scaled sharing provided by the cyberinfrastructure. Scientific workflows have recently emerged as a new paradigm for declarative representation of scientific applications as complex compositions of software components and the dataflow among them [1]. Current scientific workflow management systems, such as Pegasus [2], Kepler [3], Taverna [4], VisTrails [5] and VIEW [6], facilitate the definition and execution of scientific workflows. However, it is often very hard to create and maintain scientific workflows. In many disciplines, individual workflows are large due to the large quantities of data used. In [7] three stages in the creation of workflows are suggested to enable the management of the complexity of workflow creation by making the process more modular, but creating scientific workflows remain a challenge to domain scientists, and updating scientific workflows is also a challenge as the workflows can keep evolving when they are used by domain scientists.

Provenance, in scientific workflow community, refers to the sources of information, including entities and processes, involved in producing or delivering an artifact. Provenance is important for scientists to assess data quality, validate results, reproduce experiments, consequently provenance capture becomes an important scientific workflow research area. Many existing scientific workflow management systems, such as Taverna, Kepler and Pegasus, capture provenance information implicitly in an event log that records events related to the start and end of particular steps in the workflow execution and the corresponding data read and write events. Provenance systems above are tightly coupled with their scientific workflow management systems, while the VisTrails provenance technology [5] and infrastructure are general to enable system-level monitoring and applicable to a wide range of applications that involve complex computational processes. One of the major advantages of this general approach is that users will be able to leverage provenance using the same applications and environments that they are used to. Provenance is also argued in [8] as first class data in the cloud. We believe that complete provenance solutions will employ a combination of system-level monitoring and workflow-based systems.

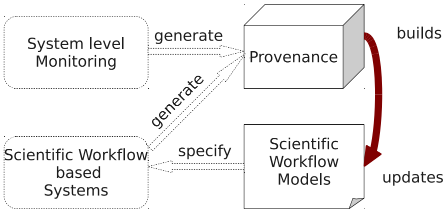


Figure 1. Mining provenance

Recent efforts from scientific workflow community [5] [9] [10] [11] [8] [12] [13] aiming at large-scale capturing of provenance present a new opportunity for building scientific workflow using provenance. Captured provenance of a single scientific experiment may come from executing existing scientific workflows, or executing provenance enabled applications. What's more, before creating scientific workflows, the provenance can only be captured from provenance enabled applications. This paper aims at answering a question: Can we learn from provenance to build scientific workflows? Several researchers have investigated how to synthesize a process model from event logs [14] [15] [16]. The research area of process mining focusses on extracting information about processes by examining event logs. Practical experience has shown that typical information recorded in event logs includes information about which activities are performed, at what time, by whom and in the context of which case (i.e., process instance) [14]. By explicitly using the case context, process discovery algorithms are capable of constructing process models that accurately describe the process [15]. Since both event logs and provenance contain process information, a given scientific workflow may be executed multiple times [17] thus creating multiple workflow execution instances. Scientific experiments are exploratory in nature thus change are the norm. As a result, mining processes from scientific workflows is highly valuable. Provenance does not record control flows associated no data flows, we are interested in building scientific workflows by combining data flows from provenance and control flows mined from provenance. Our work provides a new direction in using captured provenance.

This paper presents a method using process mining based on provenance to create and analyze scientific workflows. Figure 1 shows a high level view of the context to mine provenance. Applying process mining in the context of scientific workflow needs to address the following issues. In this paper we focus on control flow mining, and

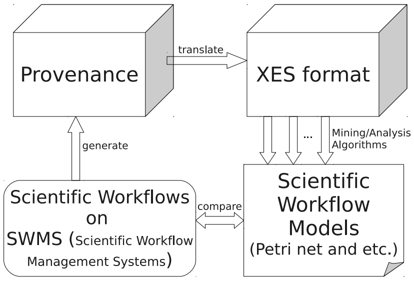


Figure 2. Overview of the method

discuss other two issues in Section IV.

1) Control flow mining: To mine control flows from provenance, we need to extract information and to present it in the format acceptable to existing process mining tools. we also need to select appropriate process discovery algorithms depending on the context of scientific workflows.
2) Data dependency: Data dependency contained in provenance can contribute to process mining for improving the mining results. It is critical to enhance the existing control flow based process mining algorithms with data flow capabilities.
3) Incremental mining: Given a scientific workflow template [7], scientists need to fine-tune it for many times, which makes updating large scientific workflows a challenge for scientists. Mining from scratch is neither efficient for large scale scientific workflows nor effective to address existing scientific workflow templates. Incremental mining can utilize the information in existing scientific workflow templates to make mining more efficient and effective.

II. A Method to Build Scientific Workflows from Provenance

Figure 1 shows a high level view of the context to mine provenance, to build and update scientific workflows. We adopt the challenge workflow from the third Provenance Challenge as a running example (http://www.myexperiment.org/workflows/750), which contains both control flow and data flow. Because there is no open provenance repository available, this paper generates provenance by running the example scientific workflow. As a result, results of the method in this paper can be compared with the existing scientific workflow. Note that the method can be applied to provenance generated from both sources in Figure 1: scientific workflows based systems and system level monitoring. Figure 2 shows a high level view of the method presented and evaluated in this paper.

A. Building Scientific Workflows through Process Discovery

1) Using the Fuzzy Miner: The fuzzy miner [16] assumes that problems in mining large scale processes are caused by mismatch between fundamental assumptions of traditional process mining, and the characteristics of real-life processes. Fuzzy miner developed an adaptive simplification and visualization technique for process models, which is based on two metrics, significance and correlation. The two metrics are similar to the concept of data clustering domain where a binary distance metric is inferred to find related subsets of attributes. In the context of scientific workflows, significance, which can be determined both for tasks and precedence relations over them, measures the relative importance of behavior. As such, it specifies the level of interest we have in tasks and their control dependency. Correlation is only relevant for precedence relations over tasks, which measures how closely related two events following one another is.

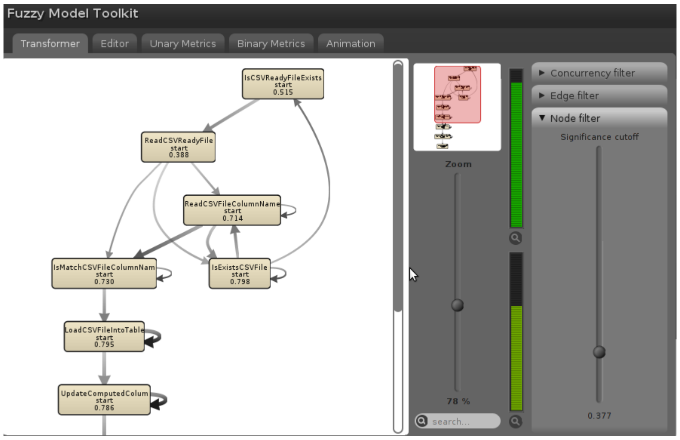


Figure 3. Fuzzy Mining Result - 1

As scientific workflows are usually quickly evolving, change can be made to the example workflow several times, including the activities and data. Using the fuzzy miner, a workflow can be mined to provide an abstract view of what does not change, which offers insight of evolving workflows. For the running example, we run it for 10 times, then remove ReadCSVReadyFile and run it for 10 times again, after that we undo removing ReadCSVReadyFile, remove IsMatchCSVFileTables and run it for 10 times. Using XESame provenance can be transformed to a XES file, based on which the fuzzy miner can be applied. Figure 3 shows a resulting model in which there is every task but IsMatchCSVFileTables, when significance cutoff is increased to 0.392, as Figure 4, ReadCSVReadyFile disappeared so that the unchanged part is shown, which can be the key part of the whole workflow. What's more, by double clicking "Cluster 14" that contains 2 elements, the tasks with low significance are shown, which in our context is the changing tasks. As Figure 5 shows, there is a process model related with low significance tasks, which exactly matchs the original workflow model in the running example. Therefore, in case there is provenance from either workflow based systems or non-workflow systems that include tasks scientists perform, a scientific workflow can be built automatically at different abstract level by using the fuzzy miner.

2) Using the Alpha Miner: The alpha miner [15] assumes the completeness of direct succession (DS) such that "if two transitions can follow each other directly, then this has occurred at least once in the log", yet it may not be the case in reality, the alpha miner allow users to edit log relations manually to offer more information about direct succession, as shown in Figure 6. For large amount of events, manually adding log relations can be impossible. In scientific workflows context, provenance contains data dependencies that imply direct succession in time order, data dependencies can somehow be considered in the alpha miner thus making it closer to completeness of direct succession. We discuss further about data dependencies in Section IV.

3) Using the Genetic Miner: The genetic miner [18] is a control-flow process mining algorithm that can discover all the common control-flow structures (i.e. sequences, choices, parallelism, loops and non-free-choices, invisible tasks and duplicate tasks) while being robust to noisy logs. The genetic miner has more difficulties to mine models with constructs that allow for many interleaving situations.

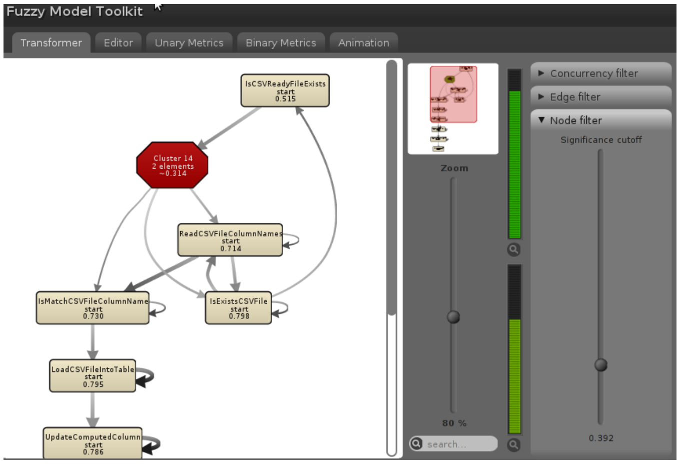


Figure 4. Fuzzy Mining Result - 2

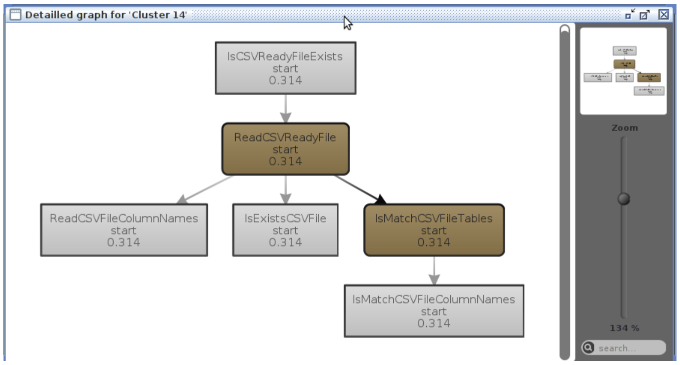
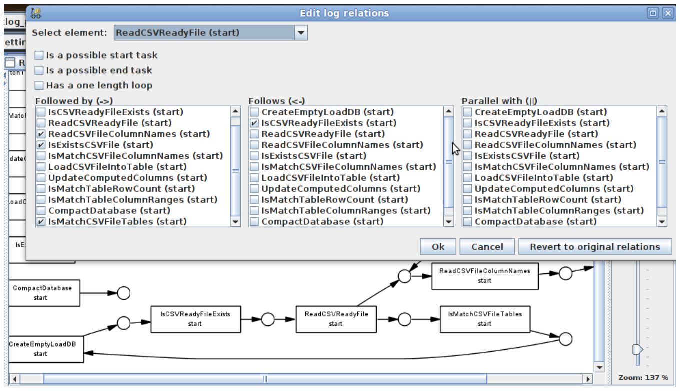


Figure 5. Fuzzy Mining Result - 3

Figure 7 shows the result of the genetic miner on the running example. Genetic miner successfully get a non-free-choices construct such as both IsMatchTableRowCount and IsMatchTableColumnRanges depend on UpdateComputedColumns while IsMatchTableRowCount depends on others as well that means mixture of choice and synchronization. It also successfully suggests the dependency between IsMatchTableRowCount and IsMatchTableColumnRanges that is a control link in the running example. The results also give a clear view of frequence by annotating numbers on each event and arc, where numbers in event boxs mean how many times the events happen in the event logs, and numbers on arcs mean how many times the two events directly succeed each other.


Edit log relations

Select element: ReadCSVReadyFile (start)

□ Is a possible start task
□ Is a possible end task
□ Has a one length loop

Followed by (->)
- □ IsCSVReadyFileExists (start)
- □ ReadCSVReadyFile (start)
- ☑ ReadCSVFileColumnNames (start)
- ☑ IsExistsCSVFile (start)
- □ IsMatchCSVFileColumnNames (start)
- □ LoadCSVFileIntoTable (start)
- □ UpdateComputedColumns (start)
- □ IsMatchTableRowCount (start)
- □ IsMatchTableColumnRanges (start)
- □ CompactDatabase (start)
- ☑ IsMatchCSVFileTables (start)

Follows (<-)
- □ CreateEmptyLoadDB (start)
- ☑ IsCSVReadyFileExists (start)
- □ ReadCSVReadyFile (start)
- □ ReadCSVFileColumnNames (start)
- □ IsExistsCSVFile (start)
- □ IsMatchCSVFileColumnNames (start)
- □ LoadCSVFileIntoTable (start)
- □ UpdateComputedColumns (start)
- □ IsMatchTableRowCount (start)
- □ IsMatchTableColumnRanges (start)
- □ CompactDatabase (start)

Parallel with (||)
- □ CreateEmptyLoadDB (start)
- □ IsCSVReadyFileExists (start)
- □ ReadCSVReadyFile (start)
- □ ReadCSVFileColumnNames (start)
- □ IsExistsCSVFile (start)
- □ IsMatchCSVFileColumnNames (start)
- □ LoadCSVFileIntoTable (start)
- □ UpdateComputedColumns (start)
- □ IsMatchTableRowCount (start)
- □ IsMatchTableColumnRanges (start)
- □ CompactDatabase (start)

Figure 6. Alpha Mining Result

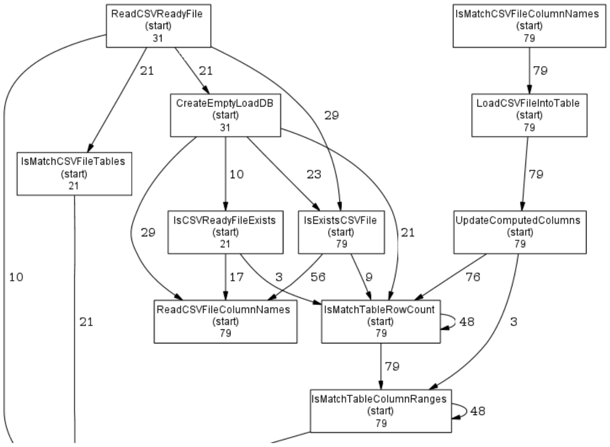


Figure 7. Genetic Mining Result

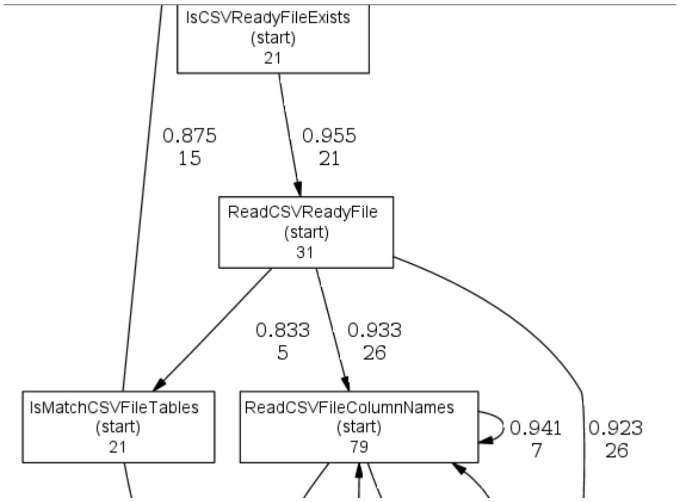
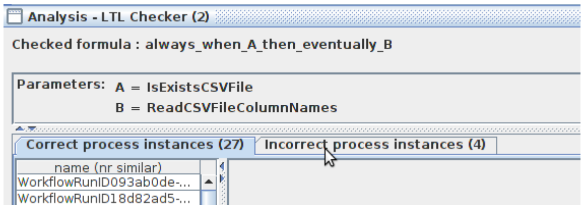


Figure 8. Heuristic Mining Result

4) Using the Heuristic Miner: The heuristics Miner [19] is a practical applicable mining algorithm that can deal with noise, and can be used to express the main behavior (i.e. not all details and exceptions) registered in an event log. It includes three steps: (1) the construction of the dependency graph, (2) for each activity, the construction of the input and output expressions and (3) the search for long distance dependency relations. Figure 8 shows the result of heuristics miner on the running example. Although IsMatchCSVFileTables does not directly succeed ReadCSVReadyFile in event logs, heuristics miner successfully suggests their dependency with reliability 0.833 and it happens 5 times in event logs considering long distance dependency relations. This is particularly useful in the context of scientific workflows, just as the running example, many scientific workflows have multiple tasks even hundreds of tasks scheduled in parallel, not each parallel task succeed the dependent task directly in provenance, therefore, long distance dependency discovery is especially important in the context of scientific workflows.

B. Analyzing Scientific Workflows Using LTL Checking

The size of provenance is growing large quickly, Linear Temporal Logical (LTL) checking is a great tool to help scientists discovering


Analysis - LTL Checker (2)
Correct process instances (27)
Incorrect process instances (4)

Checked formula : always_when_A_then_eventually_B

Parameters: A = IsExistsCSVFile
B = ReadCSVFileColumnNames

name (nr similar)
WorkflowRunID093ab0de-...
WorkflowRunID18d82ad5-...

Figure 9. LTL Checking Example

and double checking temporal properties of provenance. As shown in Figure 9, we can easily check whether ReadCSVFileColumnNames eventually happens when IsExistsCSVFile happens, it is true for 27 instances while false for 4 instances, for further information, the specific workflow run can be referred to according to workflow run identifier.

III. Related Works

The cloud computing and other technologies are changing the way we create, share and use information, which offers great benefits but also exposes us to serious new problems. Cheney et al. [20] believe that provenance will play an essential role in this revolution, providing data integrity, trustworthiness, authenticity, and availability, while offering potential benefits to information retrieval, collaboration, and scientific computation. Zhao et al. [12] address the queries from the provenance challenge workshop such as semantic reasoning which exposes the implicit links between provenance, e.g. the implicit links between provenance of studying any part of a human's body including chest, legs, arms and etc. An abstraction over the provenance information is presented by two means: one is the users' specified annotations that draw an interpretative link between tasks, and the other is the typed views that hide or expose the execution details of an iteration or a nested run, or the data lineage of a collection and its elements. Other works such as [10], [21] and [22] also address the queries from the provenance challenge workshop, however do not deal with mining processes from provenance.

IV. Discussion

A. *Results of different process discovery algorithms*

Section II-A presents results of four different process discovery algorithms on the running example. Table I discusses the results in the context of scientific workflows. Note that the result of each miner is correct based on given provenance, but providing different views of the provenance. It is found that the result of the fuzzy miner is closest to the original scientific workflow in the running example. Section IV-C discusses a possible way to improve the results in Table I.

B. *Number of Traces in Provenance*

As Figure 2 shows, this paper uses provenance from scientific workflow management systems. A question that current tools can not address is how many times should the scientific workflow be run to get enough traces. There should be a fixed point after that no more precedence relations to be discovered even given additional provenance. This paper manually find a point after that the mined results do not change significantly with additional provenance.

C. *Build Scientific Workflows using Data Dependency*

Scientific workflows include data dependency and control dependency, provenance provides data dependency besides temporal sequences. The method provided in this paper only uses the temporal sequences of tasks in provenance to mine dependency among tasks. Data dependency can contribute to process mining for improving the mining result, but process mining and its existing tools do not accept explicit data dependency as source. Since provenance provides data dependency, we can derive causality relation from data dependency, which compliments the causality relation extracted from the precedence of tasks [23].

D. *Incremental Scientific Workflow Mining*

Scientific problem solving is an evolving process. Scientists start with a set of questions then observe phenomenon, gather data, develop hypotheses, perform tests, negate or modify hypotheses, reiterate the process with various data, and finally come up with a new set of questions, theories, or laws. Often before this process can end in results, scientists will fine-tune the experiments, going through many iterations with different parameters [9]. Updating scientific workflows is hence a challenge for scientists. We believe with pre-existing scientific workflow template, created either manually or automatically through mining, we can apply process mining to update it based on new provenance obtained from either workflow based systems or non-workflow systems. We are working on incremental scientific workflow mining. Incremental mining can utilize the information in existing scientific workflow templates to make mining more efficient for large scale scientific workflows and more effective for addressing existing scientific workflow templates.

E. *Visualization of Provenance*

Provenance is typically visualized as a graph, however, the graph of a large scale workflow ususally exceeds a visually manageable size. There can be two approaches to address it. First, provenance queries can be used to focus on part of provenance. Second, the abstraction of these complex graphs can be provided to give a high level view of the provenance. However, it is difficult to write provenance query statements, and abstraction does not give details for navigation of the whole provenance. The method in this paper may be applied to improve navigation of collected provenance, by providing a zoomable view. Furthermore, the unchanged part and changed part of an evolving workflow can be highlighted. Another benefit of the method in this paper in terms of visualization, is finding relations in collaborative work scenarios to help understanding the origin of data. For example, several workflows share the access to the same data product, the method in this paper may find the relation and visualize it.

V. Conclusion

This paper provides a method using process mining to build and analyze scientific workflows, which offers a new approach to build large scale workflows in the context of scientific workflows. Recent efforts from scientific workflow community on capturing provenance present a new opportunity for using provenance. This paper presents a method using process mining based on provenance to build and analyze scientific workflows, which provides a new direction in using captured provenance. Given the fact that provenance captured in any scientific workflow based systems or system level monitoring systems contains information about tasks and their temporal order, there is always a way to translate the provenance to XES format acceptable to process mining tools, the method provided in this paper can be applied to any scientific workflow management systems.

Acknowledgement

This work was partially supported by the NSF of U.S. under award OISE-0730065, HRD-0833093, and Presidential Fellowship of Florida International University.

Table I
DISCUSSION ON RESULTS OF PROCESS DISCOVERY ALGORITHMS

	Description	Result
Fuzzy Miner	Provides a zoomable view of scientific workflows by controlling significance cutoff to show tasks at different importance level.	Under certain significance cutoff, the fuzzy miner successfully gives the changed part and unchanged part. Comparing with original scientific workflow, the fuzzy miner gets most dependency correctly, but concludes some dependency that does not exist.
Alpha Miner	Provides a view of direct sucession between tasks in provenance.	Assuming the completeness of direct succession, the alpha miner fails to give a view close to the original scientific workflow.
Genetic Miner	Provides a view of frequence for both tasks and succession between tasks, and discovers all common control-flow structures assuming the existence of noises.	The genetic miner gets a good view of structures and frequences, yet gives some wrong dependency which does not exist in both the original scientific workflow and the results of the fuzzy miner.
Heuristic Miner	Provides a view of scientific workflows by considering long distance dependency.	The heuristic miner gives long distance dependency successfully, but gives too much dependency for some tasks such as ReadCSVFileColumnNames.

REFERENCES

[1] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers, "Examining the challenges of scientific workflows," *Computer*, vol. 40, no. 12, pp. 24–32, 2007.

[2] E. Deelman, G. Singh, M. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz, "Pegasus: A framework for mapping complex scientific workflows onto distributed systems," *Sci. Program.*, vol. 13, no. 3, pp. 219–237, 2005.

[3] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao, "Scientific workflow management and the kepler system," *Concurr. Comput. : Pract. Exper.*, vol. 18, no. 10, pp. 1039–1065, 2006.

[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. Web Server issue, pp. 729–732, 2006.

[5] S. P. Callahan, J. Freire, C. E. Scheidegger, C. T. Silva, and H. T. Vo, "Towards provenance-enabling paraview," in *Provenance and Annotation of Data and Processes*, ser. Lecture Notes in Computer Science, vol. 5272. Springer Berlin / Heidelberg, 2008, pp. 120–127.

[6] A. Chebotko, C. Lin, X. Fei, Z. Lai, S. Lu, J. Hua, and F. Fotouhi, "VIEW: a VIsual sciEntificWorkflow management system," in *IEEE World Congress on Services*, 2007, pp. 207–208.

[7] Y. Gil, V. Ratnakar, E. Deelman, G. Mehta, and J. Kim, "Wings for pegasus: Creating large-scale scientific applications using semantic representations of computational workflows," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 22, 2007, p. 1767.

[8] K. Muniswamy-Reddy and M. Seltzer, "Provenance as first class cloud data," *SIGOPS Oper. Syst. Rev.*, vol. 43, no. 4, pp. 11–16, 2010.

[9] I. Altintas, O. Barney, and E. Jaeger-Frank, "Provenance collection support in the kepler scientific workflows system," *In Proceedings of the International Provenance and Annotation Workshop (IPAW'06)*, vol. 4145, p. 118, 2006.

[10] D. A. Holland, M. I. Seltzer, U. Braun, and K. Muniswamy-Reddy, "PASSing the provenance challenge," *Concurrency and Computation: Practice and Experience*, vol. 20, no. 5, pp. 531–540, 2008.

[11] J. Kim, E. Deelman, Y. Gil, G. Mehta, and V. Ratnakar, "Provenance trails in the Wings/Pegasus system," *Concurrency and Computation: Practice and Experience*, vol. 20, no. 5, pp. 587–597, 2008.

[12] J. Zhao, C. Goble, R. Stevens, and D. Turi, "Mining taverna's semantic web of provenance," *Concurrency and Computation: Practice and Experience*, vol. 20, no. 5, pp. 463–472, 2008.

[13] C. Lim, S. Lu, A. Chebotko, and F. Fotouhi, "Prospective and retrospective provenance collection in scientific workflow environments," in *Services Computing, IEEE International Conference on*, 2010, pp. 449–456.

[14] W. M. P. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 16, no. 9, p. 1128–1142, 2004.

[15] B. F. Dongen, A. K. A. D. Medeiros, and L. Wen, "Process mining: Overview and outlook of petri net discovery algorithms," in *Transactions on Petri Nets and Other Models of Concurrency II: Special Issue on Concurrency in Process-Aware Information Systems*. Springer-Verlag, 2009, pp. 225–242.

[16] C. W. Günther and W. M. P. van der Aalst, "Fuzzy mining: adaptive process simplification based on multi-perspective metrics," in *Proceedings of the 5th international conference on Business process management*, Brisbane, Australia, 2007, pp. 328–343.

[17] S. Bowers, T. McPhillips, M. Wu, and B. Ludäscher, "Project histories: managing data provenance across collection-oriented scientific workflow runs," in *Proceedings of the 4th international conference on Data integration in the life sciences*. Philadelphia, PA, USA: Springer-Verlag, 2007, pp. 122–138.

[18] A. K. Medeiros, A. J. Weijters, and W. M. P. van der Aalst, "Genetic process mining: an experimental evaluation," *Data Min. Knowl. Discov.*, vol. 14, pp. 245–304, April 2007.

[19] A. J. M. M. Weijters and A. K. A. D. Medeiros, "Process mining with the HeuristicsMiner algorithm," *Technische Universiteit Eindhoven, Tech. Rep. WP*, vol. 166, 2006.

[20] J. Cheney, S. Chong, N. Foster, M. Seltzer, and S. Vansummeren, "Provenance: a future history," in *Proceeding of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*. Orlando, Florida, USA: ACM, 2009, pp. 957–964.

[21] C. Scheidegger, D. Koop, E. Santos, H. Vo, S. Callahan, J. Freire, and C. Silva, "Tackling the provenance challenge one layer at a time," *Concurrency and Computation: Practice and Experience*, vol. 20, no. 5, pp. 473–483, 2008.

[22] S. Cohen-Boulakia, O. Biton, S. Cohen, and S. Davidson, "Addressing the provenance challenge using ZOOM," *Concurrency and Computation: Practice and Experience*, vol. 20, no. 5, pp. 497–506, 2008.

[23] R. Zeng, X. He, and W. M. P. van der Aalst, "A method to mine workflows from provenance for assisting scientific workflow composition," in *IEEE World Congress on Services*, Washington DC, USA, 2011, to appear.