Using the Provenance from Astronomical Workflows to Increase Processing Efficiency

Michael A. C. Johnson[1]*, Luc Moreau[2] Adriane Chapman[1], Poshak Gandhi[1], and Carlos Sáenz-Adán[3]**

[1] University of Southampton, Southampton, Hampshire, SO17 1BJ, UK,
{Michael.Johnson,Adriane.Chapman,Poshak.Gandhi}@soton.ac.uk
[2] King's College London, London, WC2B 4BG, UK
Luc.Moreau@kcl.ac.uk
[3] Department of Mathematics and Computer Science, University of La Rioja, Spain
carlos.saenz@unirioja.es

Abstract. Astronomy is increasingly becoming a data-driven science as the community builds larger instruments which are capable of gathering more data than previously possible. As the sizes of the datasets increase, it becomes even more important to make the most efficient use of the computational resources available. In this work, we highlight how provenance can be used to increase the computational efficiency of astronomical workflows. We describe a provenance-enabled image processing pipeline and motivate the generation of provenance with two relevant use cases. The first use case investigates the origin of an optical variation and the second is concerned with the objects used to calibrate the image. The provenance was then queried in order to evaluate the relative computational efficiency of use case evaluation, with and without the use of provenance. We find that recording the provenance of the pipeline increases the original processing time by ∼45%. However, we find that when evaluating the two identified use cases, the inclusion of provenance improves the efficiency of processing by ∼99% and ∼96% for Use Cases 1 and 2, respectively. Furthermore, we combine these results with the probability that Use Cases 1 and 2 will need to be evaluated and find a net decrease in computational processing efficiency of 13-44% when incorporating provenance generation within the workflow. However, we deduce that provenance has the potential to produce a net increase in this efficiency if more uses cases are to be considered.

1 Introduction

Provenance is a staple in the art communities as it is a record of the origin, ownership and custody of a work of art or artefact. In this context, it can be used to assess the authenticity and probe past possession, in order to value a

* MJ acknowledges support from ESPRC (EP/N509747/1)

** This work was partially supported by the spanish MINECO project EDU2016-79838-P, and by the U. of La Rioja (grant FPI-UR-2015)

work of art. The practice of provenance has also been adopted by the scientific community as reliability and reproducibility are two of its fundamental axioms. The use of provenance within science is becoming ever more important as the quantities of data and the number of people analysing each dataset increase.

Over the last few decades, the ability of the astronomer to collect and process data has increased dataset sizes from giga to tera to now peta-byte scale datasets. This is in part due to the creation of large scale survey telescopes such as the Sloan Digital Sky Survey (SDSS)[1], the Palomar Transient Factory[2] and, in future, the Large Synoptic Survey Telescope (LSST)[3]. As astronomy is increasingly becoming a data-driven science, many frameworks and tools have been designed to automate the generation of the accompanying provenance. Producing this detailed record of the provenance requires additional storage and introduces an initial runtime overhead to the execution time. However, it can also allow for a significant reduction in resources when analysing the final data products.

With the advent of new survey telescopes, such as LSST, which have extremely large datasets, it is becoming ever more crucial for the astronomer to make the most efficient use of the computational resources available. PROV-TEMPLATE[4] is a declarative approach to enable the generation of PROV compatible provenance and in this paper we investigate the implementation of PROV-templates as a means of producing the provenance of astronomical workflows. The aim is to quantitatively demonstrate the relative computational efficiency of astronomical image processing with and without the use of PROV-TEMPLATE generated provenance. In order to achieve this, firstly, PROV-TEMPLATES were used to generate the provenance of an astronomical image processing pipeline which was designed to measure the brightness variation of black hole binary systems. Secondly, within the context of this workflow, two use cases were identified for which provenance is vital for the astronomy community. Use Case 1 was to investigate the origin of an observed variation in a target astronomical objects brightness and in Use Case 2, a star was found to be incorrectly measured and it was investigated whether this star was used in the calibration process. These use cases were then evaluated with and without the use of the generated provenance and the relative resources required by each method were quantified. Finally, the total impact of provenance capture and usage was measured by comparing the computational resources required for implementation and use case evaluation with and without the use of provenance.

The contributions of this paper are: identifying two use cases for which provenance is vital for the astronomy community; a quantitative measurement of the impact of provenance capture and usage with these use cases and the application of PROV-Templates to a real world situation.

The structure of this paper is as follows, Section 2 outlines the astronomy application and identifies the use cases which will be evaluated. Section 3 describes the provenance generation method. Section 4 details the evaluation of the outlined use cases. Section 5 outlines the related work and finally, Section 6 discusses our findings.

2 Astronomy Application

The motivation of this paper is to investigate the potential for provenance to increase the efficiency of processing astronomical data, therefore we outline an astronomical dataset and image processing pipeline in this section. The astronomical images used throughout this were all taken of the low mass X-ray binary (LMXB), GS 1354-64 which consists of a star in orbit around a black hole. The pipeline identifies the objects in the images, measures the brightness of all objects and calibrates them to account for changing viewing conditions in order to find the variations in flux that GS 1354-64 exhibits over time. These optical variations can be used to determine properties of the system such as its orbital period, which can then be used, in-conjunction with spectral information, to infer the masses of the binary components. Currently, this is the only way we have to robustly measure the mass of stellar mass black holes and increasing the sample of known black hole masses enables us to better understand their properties. Survey telescopes are the ideal equipment in order to discover more systems as they are designed to systematically observe large swathes of the sky. As we are looking to discover new LMXBs, we do not know their position, although we may know areas of the sky where they are more likely to be. This means that large quantities of data must be analysed in order to find the objects of interest and it is essential to utilise any advantage in computational efficiency available which motivates our investigation into the use of provenance in this regard.

2.1 The Image Processing Pipeline

The image processing pipeline had two main functions: differential photometry and pattern recognition. As the measured brightness of the object in an image is dependent on conditions such as clouds, the image's proximity to the moon and light pollution, the images must be calibrated via differential photometry, whereby stars of known and constant brightness within the same image are used to adjust the measured brightness for differences in observing conditions. The pattern recognition was required in order to determine which source in the image corresponds to which astronomical object. The use cases are both concerned with differential photometry, therefore the explanation of the workflow will focus on this aspect.

The left hand side of Figure 1 is a UML sequence diagram depicting a simplified version of the differential photometry in the image processing pipeline. The two lifelines of the UML diagram represent the script itself and the astronomical images. The first message, *performAperturePhotometry*, measures the brightness of all objects within the image. Then, *differentialPhotometry* compares the measured brightness of known objects (standard stars) to their true brightness in order to calculate the brightness correction needed for that particular image. The pipeline determined which stars should be used as standard stars for each image individually. Multiple standard stars were used in order to get a more consistent calibration as any individual star is more effected by things such as noise or systematic uncertainties. Bright stars were also chosen for the same

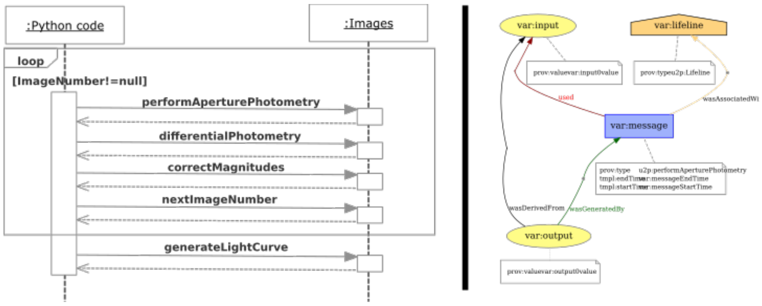


Fig. 1: The left hand side is a UML sequence diagram depicting a simplified version of the differential photometry process. The right-hand side is a PROV template generated from *performAperturePhotometry*.

reason. Once some candidate stars had been selected, they were cross-referenced with the SIMBAD astronomical database [5] to determine whether they were non-variable stars and if they were found to be so, then their true brightness was retrieved and compared to the measured value and the brightness correction for that image could be calculated. This process was repeated for each standard star in the image and the final correction was the averaged value. The brightness of the target object (in this case GS 1354-64) was then adjusted using this correction. This process was then repeated for all images. Finally, the corrected brightness of the object across all images was plotted against time to give the lightcurve, demonstrating the objects temporal optical variation.

2.2 Use Cases

In order to assess the usefulness of provenance for the astronomical community, the following use cases have been identified.

USE CASE 1. *Variation Investigation* - An Astronomer, Alice, detects a change in luminosity in a star between two images taken on two different nights. Abe *determines whether the change was intrinsic to the object or a result of the image processing pipeline.*

First, this use case requires a record of the version of the pipeline that was used for the image processing. The change in brightness could also be the result of the standard stars used to correct the measurement, either different stars being selected for each image processing step.

If the image processing is found to be consistent between the observations, then the change in observed brightness can be deduced to be due to the object, however if there are inconsistencies then the images must be reprocessed to determine the true origin of the variation.

With no accompanying provenance, the processing would have to be repeated, ensuring the pipeline was identical in order to dispel any doubt in the origin of the variation.

Evaluation of Use Case 1 asserts absolute certainty that the origin of the optical variation was not due to the image processing pipeline. However, it is usually expected, for this application, that the origin of the variation is from the object. Therefore, it is likely that Use Case 1 would only be evaluated when the astronomer, Alice, detects an unexpected result, such as too much variation or no variation at all. An unexpected result from astronomical images is not uncommon, however, quantifying how often this will occur is difficult to determine as this kind of data is typically poorly documented within the astronomical community. Consequently, estimated probabilities of 1%, 10% and 30% were all investigated in order to assess the impact of evaluating Use Case 1 on the total computational resources required.

USE CASE 2. *Calibration Propagation* - A star that was previously thought to be standard has been shown to demonstrate variability. Alice *determines which objects used this star for calibration and recalculates the photometry for them.*

Standard stars are objects of known and constant luminosity that astronomers use to calibrate images. If a standard star that was used for calibration had a different brightness than what was accounted for, then the calibration could be incorrect and an incorrect calibration means that the measured brightness of the target object is wrong, invalidating the results.

Without the use of provenance, there are two possible solutions for this calibration propagation: firstly, with no knowledge of the standard stars used for calibration, all images which contain the previously standard star would have to be re-processed, ensuring that this star is not selected; secondly, the workflow could be re-run up until the standard stars are selected from each image, and with this information, only the images which use the previously standard star in the calibration would be repeated.

Conversely, when evaluating this use case with provenance, the provenance can be queried to return the list of standard stars used in the calibration process for each image. From this, only the images which contain the newly variable star have to be re-processed.

The invalidation of the use of a standard star could also be due to an incorrectly measured brightness as well as incorrectly determining the object to be variable. Determining how often Use Case 2 is likely to be evaluated is not trivial by any means as an object may be incorrectly measured or identified if: the object saturated the image; a cosmic ray interfered with the image; there were unaccounted for artefacts or systematics; the standard object exhibited sporadic variation or it transitioned into a variable object. Taking into account all of these scenarios, an estimated 1% probability that Use Case 2 would need to be evaluated was assumed. It should be noted that this number could be calculable if provenance use was more ubiquitous within the astronomy community.

3 Provenance in Astronomy Simulations

Whilst the aim of this paper is to demonstrate the use of provenance to reduce the overall processing cost, we must also address the initial overhead introduced by provenance capture. The PROV-TEMPLATE[4] approach was used to generate PROV-compatible provenance which described the workflow. Firstly, the full pipeline was modelled as a UML Sequence Diagram and later, UML2PROV[6] was used to generate *templates* that described the design of the provenance to be generated for each function. During the execution of the workflow, *bindings* were generated every time a function was called which contained the variable-value pairs (such as inputs or outputs) that were specific to that call of the function and had corresponding variables on the template for that function. On the right-hand side of Figure 1 we can see a template generated from *performAperturePhotometry*. After completion of the workflow, these *bindings* were then expanded with their corresponding *templates* using the ProvToolbox[4] to yield the individual provenance files. These were then merged to produce the full provenance that described the system.

The image processing pipeline analysed a series of 10 images of LMXB GS 1354-64 taken by the Faulkes Telescope. All of the computation was repeated twenty times and the results in Figure 2 a) represent the average and standard deviation of these execution times. One should note that the only relevant time increase for workflow execution time is the addition of *bindings* as the merging and expansion can both be done post pipeline. The size of the products of the workflow with and without provenance were also assessed and are shown in Table 1. The size of the inputs are also included to demonstrate that whilst the provenance files are large when compared to the outputs, they are still inconsequential on the scale of the full workflow.

All simulations were run on a Dell Latitude E7470 laptop with the following specifications: 8GB of system memory; an Intel®Core™ i5-6200U CPU @ 2.30GHz. The machine was running Ubuntu 16.04, kernel: 4.4.0-112-generic.

4 Evaluation

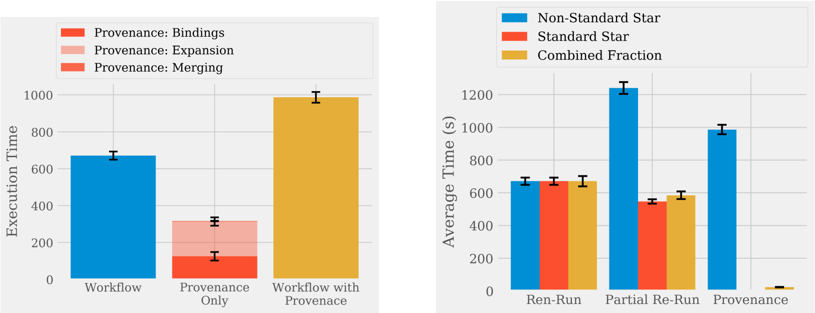
4.1 Use Case 1

The astronomical pipeline may not always perform a consistent analysis from image to image. It may have different parameters during the calibration such

[4] https://lucmoreau.github.io/ProvToolbox/

Table 1: The size of inputs consumed by and outputs produced by the image processing pipeline with and without provenance generation.

Method	Total Input Size	Total Output Size
Workflow Only	21MB	20kB
Workflow with Provenance	21MB	546kB



(a) Timing: Workflow Execution, with and without Provenance

(b) Timing: Analysis of Use Case 2

Fig. 2: a) Average processing times for workflow execution, with and without provenance generation. b) Computational resources required to evaluate Use Case 2, when implementing different solutions. Execution times vary depending on whether the newly variable star was used as a standard star in the calibration on not, so both times are shown. The combined fraction convolves these processing times with the probability that any star in the image was used as a standard star. Both sets of results are the average found over twenty simulations and the error bars represent their standard deviation.

as which stars were used as standard stars. It may also use different library versions of the pipeline and the path that each data product made through the pipeline may not always be the same. Use Case 1 investigates an observed change in brightness from one image to another and tries to determine whether this variation was inherent to the object itself or whether its origin was due to inconsistencies in the image processing pipeline.

In order to evaluate Use Case 1 without provenance, the workflow must be rerun over the series of images where the variation was observed, with the pipeline versions and calibration settings made certain to be the same throughout. To evaluate Use Case 1 with the use of provenance, SPARQL queries were written to determine which versions of the pipeline and which standard stars were used for each image. The queries were < 10 lines long and had a negligible run time (< 1 second).

It was found that the same standard stars were used throughout the series of images and the versions of the pipeline used were the same throughout as well. Therefore, the observed variation could be deduced to not be due to the image processing and the data did not need to be reprocessed. This information resulted in a $\sim$99% increase in computational efficiency over evaluating the use case without provenance. Table 2 shows the processing time necessary for evaluating each use case, as well as the length of the code required to do so.

4.2 Use Case 2

Use Case 2 was to determine whether a star that was recently determined to be variable was used in the image processing as a standard star and therefore invalidated the calibration for that image. Three ways of evaluating Use Case 2 were investigated: firstly, the workflow was completely re-executed, ensuring that the variable star is not used in the calibration process; secondly the workflow was executed up until the selection of standard stars, this information recorded and the images which contain the variable object were re-computed and finally, the provenance of the workflow was queried to determine which images should be re-processed.

For the first case, the time to evaluate Use Case 2 is the same as the original execution time as there is no information on which images did or did not use the variable object for calibration so all must be repeated. For the second scenario, the evaluation time is reduced when the variable star was found not to be used as a standard star as the workflow had to only be partially re-run. However, if it were found to be used as a standard star then the workflow must also be completely re-run with this star not being used in the calibration in addition to the partial run to find the standard stars used. The third evaluation queries the provenance in order to determine whether the newly variable star was used as a standard star. In summary, the first evaluation assumes no knowledge of the workflow and always completely re-runs. The second method determines information on the standard stars used by partially re-running the workflow then deciding whether it should all be re-run. The final method leverages provenance information in order to determine whether the workflow should be re-run.

If it was not used as a standard star, then there was only the computational cost of provenance querying required to evaluate the use case as the workflow does not need to be re-run. If it was used as a standard star, then the workflow must be re-run with the newly variable star not used during the calibration process. The SPARQL queries used to evaluate this use case were < 10 lines long and had a negligible run time (< 1 second), as before.

As the computational efficiency of two of the methods rely on whether the newly variable star was used as a standard star, the probability that any star in the image was used in the calibration as a standard star was calculated. This probability was convolved with the computation time required by each method of use case evaluation for if the star was used as a standard star and if it was not. This probability, P, was defined as $P = n/A$ where n is the number of standard stars per image and A is the average number of objects per image. For this example, 10 standard stars were used and the total number of objects in the image was $\sim$450, therefore, assuming all objects were treated equally, there was a $\sim$2% chance that any star in the image was used as a standard star. By

Table 2: Computational resources required to evaluate Use Case 1, including the average run time and an order of magnitude of the lines of code needed to evaluate the use case with and without the use of provenance.

Method	Use Case Analysis Computation Time (s)	SD (s)	Lines of Code (Approximate)
Workflow Only	671	22	500
With Provenance	1	0	10

combining this probability with the two timings, we compute the average cost of use case evaluation if any given star in the image was found to be variable.

Figure 2 b) shows the results for evaluating Use Case 2 with the three possible solutions. The time represents the average execution time after repeating the simulation twenty times. The columns in Figure 2 b) represent time taken when the object found to be variable was used as a standard star, when it was not used as a standard star and both these results combined with the probability that any star in the image was used as a standard star (the combined fraction).

We found that the computational processing cost of Use Case 2 evaluation if the star is found to be standard decreases by 21% with provenance when compared to partially re-running the workflow. However, we also found that the processing time increases in this respect with the use of provenance by 47% when compared to simply re-running the workflow. This is due to three reasons: firstly, the initial overhead of provenance production; secondly the relatively small cost of querying the provenance and finally, the workflow must be completely re-run in either case as the fact the star was used as a standard star invalidates the initial results. We also found that the cost of evaluation is greatly reduced if the star was not used as a standard star because here, the only computational cost is for querying the provenance which is negligible when compared to re-running the workflow, increasing the efficiency by ∼99% when compared to either evaluation without the use of provenance. Finally, when we combine these efficiencies with the probability that the star will be used as a standard star, we found that with the use of provenance, the computational efficiency of evaluating Use Case 2 increases by a factor of 97% and 96% when compared to evaluating it by re-running and partially re-running the workflow, respectively.

5 Related Work

5.1 Provenance in Astronomy and e-Science

An early example of provenance within e-Science was outlined in Lanter et al. [7] where they designed lineage meta-data base system in order to document the sources of data in geographic information system (GIS) applications. This information then assisted in determining the quality of the data and the fitness of use for potential applications. Another example framework is $^{\text{my}}$Grid[8], designed to meet the needs of *in silico* experiments in biology. $^{\text{my}}$Grid prioritises semantic complexity over availability of computationally intensive resources to reflect the data centric nature of the bioinformatic experiments.

Examples of frameworks designed with the needs of the astronomy community in mind are Chimera[9] and Kepler[10]. One of the motivations for Chimera was SDSS, their data intensive needs and the requirement for scalability. Chimera therefore developed the virtual data system which allowed for on demand data generation, reducing the storage requirements.

Many scientists have adopted the use of scripting languages rather than working within scientific workflow systems due to their relative proficiency in them. Fortunately for the modern astronomer, tools such as YesWorkflow[11] and NoWorkflow[12] have been developed to automate the generation of provenance from these scripts. P.Groth et al.[13] explored the use of provenance queries within astronomy. They identified relevant astronomy use cases for provenance which motivated the construction of a new provenance model which requires less storage than traditional provenance generation in anticipation for the large data production expected by LSST.

5.2 PROV-TEMPLATES

PROV-TEMPLATES facilitate the design and generation of provenance compatible with the PROV standard of the world wide web consortium[4]. PROV-TEMPLATE generated provenance has previously been employed by A Giesler et al. [14] to provide provenance tracking in scientific workflows.

One advantage of PROV-TEMPLATES over other methods of provenance generation is that only the bindings need be created during workflow execution and they can then be expanded later. This not only reduces the initial processing required at execution, but also can reduce the storage requirement as the bindings are typically only 40% of the size of the expanded provenance templates[4]. PROV-TEMPLATES also facilitate the generation of provenance without the need for writing code to do so such as the tools YesWorkflow[11] or NoWorkflow[12]. However, unlike these systems, PROV-TEMPLATES also allow for complex queries over the provenance that are possible in purpose built frameworks such as Chimera[9].

6 Conclusions

We have found that recording the provenance of an image processing pipeline increases the initial processing cost by ∼45%. However, we have also demonstrated that the use of provenance resulted in an increase in computational efficiency of 99% and 96% when evaluating Use Cases 1 and 2, respectively. We speculated that evaluation of Use Case 1 would occur from 1% to 30% of the time and Use Case 2 would likely need to be evaluated ∼1% of the time. By combining the processing cost of provenance production, use case evaluation and the probability that the use cases will need to be evaluated, we compute the total net change in processing efficiency of the workflow by introducing provenance generation as a decrease in computational processing efficiency of 13-44%, depending on how often Use Case 1 needs to be evaluated. The full results are shown in Table 3.

We also found that when including provenance, the total size of artefacts produced by the workflow increased by a factor of ∼6. Whilst these results do represent a large increase in data products, it should be noted that they are completely un-optimised for storage space savings. Also the provenance is fairly fine-grained and has the potential to evaluate many other use cases not investigated in this paper. This means that there is the possibility for a significant reduction in both the size of the final provenance and its intermediate products.

Furthermore, the combined data products from provenance production and the workflow still represented $< 1\%$ of the total data products consumed by the pipeline as the size of the input images dwarfs that of the data products.

These results pertain to the image processing pipeline used during this paper and it is likely to change from pipeline to pipeline. Having said this, other pipelines which are designed to achieve the same goals will likely be similar in operation and correlate with the results found in this paper. One interesting investigation would be the comparison between results obtained with the use of PROV standard provenance vs the home-grown provenance solutions developed by astronomers as part of their scripts.

One limitation of our approach was determining the probability that the use cases would need to be evaluated as we were only able to postulate estimated probabilities. The more often these use cases need to be evaluated, the more provenance positively impacts the computational efficiency of the workflow. The results therefore only serve as an estimation of the impact of provenance recording on the computational efficiency of astronomical workflows.

The results suggest that implementing provenance recording on astronomical workflows has a negative impact on the computational resources required. However, it has been clearly demonstrated that including provenance vastly reduces the evaluation time of the outlined use cases and identifying more use cases would therefore increase net computational efficiency of the workflow when using provenance.

In conclusion; can provenance be used to decrease the computational resources consumed by astronomical workflows? No, if the only use cases for provenance are the the two outlined in this paper. However, there is the potential to do so with additional investigation into use cases for astronomical provenance.

References

1. Donald G York, J Adelman, John E Anderson Jr, Scott F Anderson, James Annis, Neta A Bahcall, JA Bakken, Robert Barkhouser, Steven Bastian, Eileen Berman, et al. The sloan digital sky survey: Technical summary. *The Astronomical Journal*, 120(3):1579, 2000.
2. Nicholas M Law, Shrinivas R Kulkarni, Richard G Dekany, Eran O Ofek, Robert M Quimby, Peter E Nugent, Jason Surace, Carl C Grillmair, Joshua S Bloom,

Table 3: Total computational processing cost of running the workflow with and without provenance. Including processing cost of use case analysis combined with the probability that the use case must be evaluated. Use Case 1 results are combined with the probability the use case would need to be evaluated 1%, 10% and 30% of the time.

	Workflow Run Time (s)	Use Case 1 Run Time (s)	Use Case 2 Run Time (s)	Total Run Time (s)
		(1%,10%,30%)		(1%,10%,30%)
Workflow Only	671	7, 67, 201	6	684, 744, 878
Workflow with Provenance	987	<1	<1	988

Mansi M Kasliwal, et al. The palomar transient factory: system overview, performance, and first results. *Publications of the Astronomical Society of the Pacific*, 121(886):1395, 2009.

3. J Anthony Tyson. Large synoptic survey telescope: overview. In *Survey and Other Telescope Technologies and Discoveries*, volume 4836, pages 10–21. International Society for Optics and Photonics, 2002.
4. Luc Moreau, Belfrit Batlajery, Trung Dong Huynh, Danius Michaelides, and Heather Packer. A templating system to generate provenance. *IEEE Transactions on Software Engineering*, 2017.
5. Marc Wenger, François Ochsenbein, Daniel Egret, Pascal Dubois, François Bonnarel, Suzanne Borde, Françoise Genova, Gérard Jasniewicz, Suzanne Laloë, Soizick Lesteven, et al. The simbad astronomical database-the cds reference database for astronomical objects. *Astronomy and Astrophysics Supplement Series*, 143(1):9–22, 2000.
6. Carlos Sáenz-Adán, Beatriz Pérez, Trung Dong Huynh, and Luc Moreau. UML2PROV: Automating provenance capture in software engineering. In *International Conference on Current Trends in Theory and Practice of Informatics*, pages 667–681. Springer, 2018.
7. David P Lanter. Design of a lineage-based meta-data base for gis. *Cartography and Geographic Information Systems*, 18(4):255–261, 1991.
8. Robert D Stevens, Alan J Robinson, and Carole A Goble. mygrid: personalised bioinformatics on the information grid. *Bioinformatics*, 19(suppl_1):i302–i304, 2003.
9. Ian Foster, Jens Vockler, Michael Wilde, and Yong Zhao. Chimera: A virtual data system for representing, querying, and automating data derivation. In *Scientific and Statistical Database Management, 2002. Proceedings. 14th International Conference on*, pages 37–46. IEEE, 2002.
10. Bertram Ludäscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A Lee, Jing Tao, and Yang Zhao. Scientific workflow management and the kepler system. *Concurrency and Computation: Practice and Experience*, 18(10):1039–1065, 2006.
11. Timothy McPhillips, Tianhong Song, Tyler Kolisnik, Steve Aulenbach, Khalid Belhajjame, Kyle Bocinsky, Yang Cao, Fernando Chirigati, Saumen Dey, Juliana Freire, et al. Yesworkflow: a user-oriented, language-independent tool for recovering workflow information from scripts. *arXiv preprint arXiv:1502.02403*, 2015.
12. Leonardo Murta, Vanessa Braganholo, Fernando Chirigati, David Koop, and Juliana Freire. noworkflow: capturing and analyzing provenance of scripts. In *International Provenance and Annotation Workshop*, pages 71–83. Springer, 2014.
13. Paul Groth, Ewa Deelman, Gideon Juve, Gaurang Mehta, and Bruce Berriman. Pipeline-centric provenance model. In *Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science*, page 4. ACM, 2009.
14. André Giesler, Myriam Czekala, Björn Hagemeier, and Richard Grunzke. Uniprov: A flexible provenance tracking system for unicore. In *Jülich Aachen Research Alliance (JARA) High-Performance Computing Symposium*, pages 233–242. Springer, 2016.