Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Common motifs in scientific workflows: An empirical analysis

Daniel Garijo [a,*,1], Pinar Alper [b,1], Khalid Belhajjame [b], Oscar Corcho [a], Yolanda Gil [c], Carole Goble [b]

[a] Ontology Engineering Group, Universidad Politécnica de Madrid, Spain

[b] School of Computer Science, University of Manchester, United Kingdom

[c] Information Sciences Institute, Department of Computer Science, University of Southern California, United States

HIGHLIGHTS

- We present an empirical analysis performed over 260 scientific workflow descriptions.
- We define a catalog of domain independent abstractions for workflows.
- We discuss the distribution of the abstractions across different workflow systems.
- Different workflow systems share a common core of workflow abstractions.
- Data preparation is an obstacle for workflow understandability.

ARTICLE INFO

Article history:
Received 1 February 2013
Received in revised form
2 August 2013
Accepted 5 September 2013
Available online xxxx

Keywords:
Scientific workflows
Workflow motif
Workflow pattern
Taverna
Wings
Galaxy
Vistrails

ABSTRACT

Workflow technology continues to play an important role as a means for specifying and enacting computational experiments in modern science. Reusing and re-purposing workflows allow scientists to do new experiments faster, since the workflows capture useful expertise from others. As workflow libraries grow, scientists face the challenge of finding workflows appropriate for their task, understanding what each workflow does, and reusing relevant portions of a given workflow. We believe that workflows would be easier to understand and reuse if high-level views (abstractions) of their activities were available in workflow libraries. As a first step towards obtaining these abstractions, we report in this paper on the results of a manual analysis performed over a set of real-world scientific workflows from Taverna, Wings, Galaxy and Vistrails. Our analysis has resulted in a set of *scientific workflow motifs* that outline (i) the kinds of data-intensive activities that are observed in workflows (*Data-Operation motifs*), and (ii) the different manners in which activities are implemented within workflows (*Workflow-Oriented motifs*). These motifs are helpful to identify the functionality of the steps in a given workflow, to develop best practices for workflow design, and to develop approaches for automated generation of workflow abstractions.

1. Introduction

A scientific workflow is a template defining the set of tasks needed to carry out a computational experiment [1]. Scientific workflows have been increasingly used in the last decade as an instrument for data intensive science. Workflows serve a dual function: first, as detailed documentation of the scientific method used for an experiment (i.e. the input sources and processing steps taken for the derivation of a certain data item), and second, as re-usable, executable artifacts for data-intensive analysis. Scientific workflows are composed of a variety of data manipulation activities such as data movement, data transformation, data analysis and data visualization to serve the goals of the scientific study. The composition is done through the constructs made available by the workflow system used, and is largely shaped by the function undertaken by the workflow and the environment in which the system operates.

A variety of workflow systems, both open source (e.g. Taverna [2], Wings [3], Galaxy [4], Vistrails [5], Kepler [6], ASKALON [7]) and commercial (e.g. Pipeline Pilot[2]) are in use in a variety of scientific disciplines such as genomics, astronomy, cheminformatics, etc. A workflow is a software artifact, and once developed and tested, it

* Corresponding author. Tel.: +34 667949892.
E-mail addresses: dgarijo@fi.upm.es (D. Garijo), alperp@cs.manchester.ac.uk (P. Alper), khalidb@cs.manchester.ac.uk (K. Belhajjame), ocorcho@fi.upm.es (O. Corcho), gil@isi.edu (Y. Gil), carole.goble@cs.manchester.ac.uk (C. Goble).

[1] The first and second authors have contributed equally to the work presented in this paper.

[2] http://accelrys.com/products/pipeline-pilot/.

0167-739X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.future.2013.09.018

can be shared and exchanged between scientists. Other scientists can then reuse existing workflows in their experiments, e.g., as sub-workflows [8].

Workflow reuse presents several advantages [9]: allowing for principled attribution of established methods, improving quality through incremental/evolutionary workflow development (by leveraging the expertise of previous users), and making scientific processes more efficient. Users can also re-purpose existing workflows to adapt them to their needs [9]. Emerging workflow repositories such as myExperiment [10] and CrowdLabs [11] have made publishing and finding workflows easier, but scientists still face the challenges of understanding and reusing the available workflows.

A major difficulty in understanding workflows is their complex nature. A workflow may contain several scientifically-significant analysis steps, combined with other data preparation or result delivery activities, and in different implementation styles depending on the environment and context in which the workflow is executed. This difficulty in understanding stands in the way of reusing workflows.

Through an analysis of the current practices in scientific workflow development, we pursue the following objectives:

1. To reverse-engineer the set of current practices in workflow development through an empirical analysis.
2. To identify workflow abstractions that would facilitate understandability and therefore effective reuse.
3. To detect potential information sources that can be used to inform the development of tools for creating workflow abstractions.

In this paper we present the result of an empirical analysis performed over 260 workflow descriptions from Taverna [2], Wings [3], Galaxy [4] and Vistrails [5]. Based on this analysis, we propose a catalog of domain independent conceptual abstractions for workflow steps that we call *scientific workflow motifs*. Motifs are provided through (i) a characterization of the kinds of data-operation activities that are carried out within workflows, which we refer to as *Data-Operation motifs*, and (ii) a characterization of the different manners in which those activity motifs are realized/implemented within workflows, which we refer to as *Workflow-Oriented motifs*.

This paper extends our previous work [12], which performed an analysis of 177 workflows from Wings and Taverna. The new contributions reported on in this paper are an extension of the related work in Section 2, the addition and extension of scientific domains from Wings and Taverna workflows (Social Network Analysis, Astronomy and Domain Independent) in Sections 3 and 5; and the analysis of workflows from the Galaxy and Vistrails systems among different domains (Genomics, Text Mining, Domain Independent and Medical Informatics). Finally, we have also revisited the motif catalog (Section 4), our previous results (Section 5) and conclusions (Section 7) according to our new findings.

2. Related work

Our motifs can be seen as higher-level patterns observed in scientific workflows. "Workflow patterns" have been extensively studied [13], where inventories of possible patterns are developed based on workflow constructs that are possible in different languages, along with the ways to combine those constructs. Scientific workflows typically use a dataflow paradigm rather than a control flow paradigm that is more typical of business workflows [14], and generic data-intensive usage patterns[3] are described in [15]. Other classifications are based on the intrinsic properties of the workflows (size, structure, branching factor, resource usage, etc.) [16,17] and their environmental characteristics (makespan, speed-up, success rate, etc.) [17]. Rather than specifying what is theoretically possible with the given constructs, our work is instead based on an empirical analysis to detect similar data-intensive activities that recur in workflows across different domains and workflow systems. In addition, our work offers a complementary perspective in that we aim to understand groupings of workflow steps that form a meaningful high-level data manipulation operation.

In Software Engineering, the term "pattern" refers to established best practices to solve recurring problems. In this regard patterns represent good and exemplary practice. In [18] the authors outline anti-patterns in scientific workflows, namely redundancy and structural conflicts. The authors go on to provide a solution to address the redundancy anti-pattern. Particularly due to this perception of the term "pattern", in this paper we opted to use the term "motif" for our classification of tasks. Our objective is to take a snapshot of the existing set of activities in workflows, rather than try to prescribe a best practice.

Our Data-Operation motifs can be seen as a domain-independent classification of tasks within scientific workflows. Similar analyses have been done in a domain-specific manner in areas such as bioinformatics, based on user studies [19]. Combined with such-domain specific classifications, motifs can make way for specification of abstract workflow templates, which can be elaborated to concrete workflows prior to their execution [20].

Another work, somewhat closer to our study in spirit, is an automated analysis of workflow scripts from the Life Science domain [21]. This work aims to deduce the frequency of different kinds of technical ways of realizing workflow steps (e.g. service invocations, local "scientist-developed" scripting, local "ready-made" scripts, etc.). [21] also drills down into the category of local ready-made scripts, to outline a functional breakdown of their activity categories such as data access or data transformation. While this provides an insight into the kind of activities undertaken in workflows, it focuses on characterizing local task types. Our approach is different from this work as we focus on detecting multi-step activities with many realizations (not just individual steps).

[22] extends the categories defined in [21] identifying subcategories at a processor level by analyzing 898 workflows in myExperiment. The main difference with our analysis is that some of the proposed categories are based on technological features of the processors (i.e., the type of script) for highlighting workflow reuse among the dataset, while our catalog relies on their functional characteristics.

Finally, Problem Solving Methods (PSMs) is another area of related work. PSMs describe the reasoning process to achieve the goal of a task in an implementation and domain-independent manner [23]. Some libraries aim to model the common processes in scientific domains [24], which could be further refined with the motifs proposed in this work.

3. Analysis setup

For the purposes of the analysis, we used workflows from Taverna [2], Wings [3], Galaxy [4] and Vistrails [5]. These systems have different features:

- Taverna [2] can operate in different execution environments and provides several possibilities of deployment. Taverna is available as a workbench,[4] which embodies a desktop design

[3] http://www.workflowpatterns.com/patterns/data/.

[4] Taverna Workbench http://www.taverna.org.uk/download/workbench/.

Please cite this article in press as: D. Garijo, et al., Common motifs in scientific workflows: An empirical analysis, Future Generation Computer Systems (2013), http://dx.doi.org/10.1016/j.future.2013.09.018

Table 1
Summary of the main differences in the features of each workflow system: explicit support for control constructs (conditional, loops), whether the user interface is web-based or not, whether the environment is open or controlled and the engine used.

Workflow system	Control constructs	GUI	Environment type	Engine
Taverna	NO	Desktop/Web	Open/Controlled	Taverna
Wings	NO	Web	Controlled	Pegasus/ Apache OODT
Galaxy	NO	Web	Open/Controlled	Galaxy
Vistrails	YES	Desktop/Web	Open/Controlled	Vistrails

UI and an execution engine. Taverna also allows standalone deployments of its engine[5] in order to serve multiple clients. In its default configuration Taverna does not prescribe that the datasets and tools are integrated into an execution environment. In this sense it adopts an open-world approach, where workflows integrate (typically) remote third party resources and compose them into data-intensive pipelines. On the other hand, it also allows the development of plug-ins for the access and usage of dedicated computing infrastructures (e.g. grids) or local tools and executables. Its use has been extended from Bioinformatics to several domains including Astronomy, Chemistry, Text Mining and Image Analysis.

- Wings [3] uses semantic representations to describe the constraints of the data and computational steps in the workflow. Wings can reason about these constraints, propagating them through the workflow structure and use them to validate workflows. It has been used in different domains, ranging from Life Sciences to Text Analytics and Geosciences. Wings provides a web based access and can run workflows locally, or submit[6] them to the Pegasus/Condor [25] or Apache OODT [26] execution environments that can handle large-scale distributed data and computations.
- Galaxy [27] is a web-based platform for data intensive biomedical research which has many followers in the scientific community [28]. One of the main features of Galaxy is its cloud backend, which provides support for its extensive catalog of tools. These tools allow performing different types of analysis of data from widely used existing datasets in the biomedical domain. Galaxy uses its own engine for managing the workflow execution, compatible with batch systems or Sun Grid Engine (SGE).[7] Galaxy workflows can be run online[8] or by setting up a local instance.[9]
- Vistrails [29] tracks the change-based provenance in workflow specifications in order to facilitate reuse. It has been used in different domains of Life Sciences like Medical Informatics and Biomedicine, but also in other domains like Image Processing, Climatology and Physics. Vistrails uses its own engine to manage the execution, which allows for the combination of specialized libraries, grid and web services. Its workflows can be run online[10] or locally.[11]

The choice of these systems was due to the availability of a pool of shared workflows through repositories [11] [10] [30] and portals[12,13] but also because of their similarities:

- They provide similar workflow modeling constructs. Unlike other workflow systems (e.g., business workflows), the selected workflow systems are often data-flow oriented, and they operate on large and heterogeneous data that may need to be archived to be used in further experiments.
- All of them are open-source scientific workflow systems, initially focused on performing in-silico experimentation on the Life Sciences (Taverna, Galaxy, Vistrails) and Geosciences (Wings) domains. Taverna, Wings and Vistrails now also have workflows across other different domains like Astronomy, Machine Learning, Meteorology, etc.
- All the systems can interact with third party tools and services, and they include a catalog of components for performing different operations with data.

It is worth mentioning that despite being similar, we can find some differences among the selected systems. In particular, Vistrails has explicit mechanisms for the basic control constructs (e.g., conditionals and looping) in one of its latest releases, while Taverna, Wings and Galaxy are observers of the pure data-flow paradigm (although there are implicit ways of implementing such control structures).

The variety of environments in which these systems operate highlights some other differences as well. While Taverna allows users to specify workflows that make use of autonomous third party services (i.e. an open environment), Wings requires that the resources and the analysis operations are made part of its environment prior to being used in experiments (i.e. controlled environment). However, the difference between the systems is not a significant differentiating factor, as Taverna allows more control to be added to the environment through the addition of plug-ins, and Wings can establish a connection to third party services via custom components. Vistrails and Galaxy could be positioned at an intermediate point, since they provide access to external web-services but also build on a comprehensive library of components.

A summary of the commonalities and differences among the workflow systems included in the analysis can be seen in Table 1.

3.1. Description of the sets of workflows analyzed

For our analysis, we have chosen 260 heterogeneous workflows in a variety of domains. We analyzed a set of public Wings workflows (89 out of 132 workflows), part of the Taverna set (125 out of 874 Taverna2 workflows in myExperiment), a set of Galaxy workflows (26 out 145 of workflows) and part of the Vistrails set (20 out 274 of workflows).

- For Wings, we have analyzed all workflows from Drug Discovery, Text Mining, Domain Independent, Genomics and Social Network Analysis domains.
- For Taverna we have analyzed workflows that were available in myExperiment [10]. We determined the groups/domains of workflows by browsing the myExperiment group tags[14] and identifying those domains which contained workflows that were publicly accessible at the time of the analysis. For the Taverna dataset we analyzed Cheminformatics, Genomics, Astronomy, Biodiversity, Geo-Informatics and Text Mining domains.

5 Taverna Server http://www.taverna.org.uk/download/server/.
6 http://www.wings-workflows.org.
7 http://star.mit.edu/cluster/docs/0.93.3/guides/sge.html.
8 https://main.g2.bx.psu.edu/root.
9 http://wiki.galaxyproject.org/Admin/Get%20Galaxy.
10 http://www.crowdlabs.org/vistrails/.
11 http://www.vistrails.org/index.php/Downloads.
12 https://main.g2.bx.psu.edu/.
13 http://www.opmw.org/sparql.
14 http://www.myexperiment.org/groups.

Please cite this article in press as: D. Garijo, et al., Common motifs in scientific workflows: An empirical analysis, Future Generation Computer Systems (2013), http://dx.doi.org/10.1016/j.future.2013.09.018

Table 2
Number of workflows analyzed from Taverna (T), Wings (W), Galaxy (G), Vistrails (V).

Domain	No. of workflows	Source			
		T	W	G	V
Drug discovery	7	0	7	0	0
Astronomy	51	51	0	0	0
Biodiversity	12	12	0	0	0
Cheminformatics	7	7	0	0	0
Genomics	90	38	28	23	1
Geo-informatics	6	6	0	0	0
Text analysis	45	11	31	3	0
Social network analysis	5	0	5	0	0
Medical informatics	7	0	0	0	7
Domain independent	30	0	18	0	12
Total	260	125	89	26	20

The distribution of workflows to domains is not even, as it is also the case in myExperiment. Taverna is the workflow system with the largest public collection of workflows, in order to obtain a feasible subset of workflows for manual analysis, we made random selections from each identified domain.

- For Galaxy we have chosen the documented workflows available in the public workflow repository[15] (i.e., those workflows with annotations explaining the functionality of their components). Since Galaxy is specialized in the biomedical domain, most of the workflows are from the Genomics domain, although some of them (3) do text analysis operations in files.
- For Vistrails we have chosen a set of documented workflows available in Crowdlabs and tutorials, which include domains in Medical Informatics and Genomics. It is worth mentioning that some workflows are domain independent (machine learning workflows, visualization and rendering of datasets, annotation of texts, etc.), so they have been included under a new category.

When selecting the workflows for the analysis we paid attention to including workflows that are developed with the intention of backing actual data-intensive scientific investigations. We refrained from including toy or example workflows, which are used for demonstrating the capabilities of different workflow systems. Table 3 provides additional information on the size of workflows analyzed in terms of the range and average number of analysis tasks. The number of workflows analyzed from each domain can be seen in Table 2.

3.2. Approach for workflow analysis

Our analysis has been performed based on the documentation, metadata and traces available for each of the workflows within the cohort studied. Each workflow has been inspected using the associated workbench/design environment. Documentation on workflows is provided within workflow descriptions and in repositories in which the workflows are published. We have performed a *bottom-up and manual* analysis that aimed to surface two orthogonal dimensions regarding activities/operations that make-up workflows: (1) outline what kind of data-operation activity is being undertaken by a workflow step and (2) how that activity has been realized. For example, a visualization step (data oriented activity) can be realized in different ways: via a stateful multi-step invocation, through a single stateless invocation (depending on the environmental constraints and nature of the services), or via a sub-workflow.

Table 3
Maximum, minimum and average number of steps within workflows per domain.

Domain	Max. size	Min. size	Avg steps
Drug discovery	18	1	7
Astronomy	33	1	7
Biodiversity	12	1	4
Cheminformatics	20	1	9
Genomics	53	1	8
Geo-informatics	14	3	8
Text analysis	15	1	5
Social network analysis	7	3	6
Medical informatics	29	8	14
Domain independent	20	1	6

Table 4
Scientific workflow motifs.

- Data-Operation motifs
  - Data preparation
    - Combine
    - Filter
    - Format transformation
    - Input augmentation
    - Output extraction
    - Group
    - Sort
    - Split
  - Data analysis
  - Data cleaning
  - Data movement
  - Data retrieval
  - Data visualization
- Workflow-Oriented motifs
  - Inter workflow motifs
    - Atomic workflows
    - Composite workflows
    - Workflow overloading
  - Intra workflow motifs
    - Internal macros
    - Human interactions
    - Stateful (asynchronous) invocations

The only automated part of the data collection was associated to the workflow size statistics for Taverna workflows. The myExperiment repository provides a REST API[16] that allows retrieving information on Taverna workflows. Using this facility we were able to automate the collection of partial statistics regarding the number of workflow steps and the number of input/output parameters of Taverna workflows.

Rather than hypothesizing possible motifs up front, we built up a set of motifs as we progressed with the analysis. For each workflow we recorded the number of occurrences of each motif (independently of the workflow system it belonged to). In order to minimize misinterpretation and human error, the motif occurrences identified for each workflow have been cross-validated by another author and discussed until agreement.

4. Scientific workflow motif catalog for abstracting workflows

This section introduces details on the scientific workflow motifs detected in our analysis. Motifs are divided into two categories: Section 4.1 introduces the Data-Operation motifs (i.e., the motifs related to the main functionality undertaken by the steps of the workflow), while Section 4.2 explains the Workflow-Oriented motifs (i.e., how the Data-Operation motifs are undertaken in the workflow). An overview is provided in Table 4.

15 https://main.g2.bx.psu.edu/workflow/list_published.

16 http://wiki.myexperiment.org/index.php/Developer:API.

Please cite this article in press as: D. Garijo, et al., Common motifs in scientific workflows: An empirical analysis, Future Generation Computer Systems (2013), http://dx.doi.org/10.1016/j.future.2013.09.018

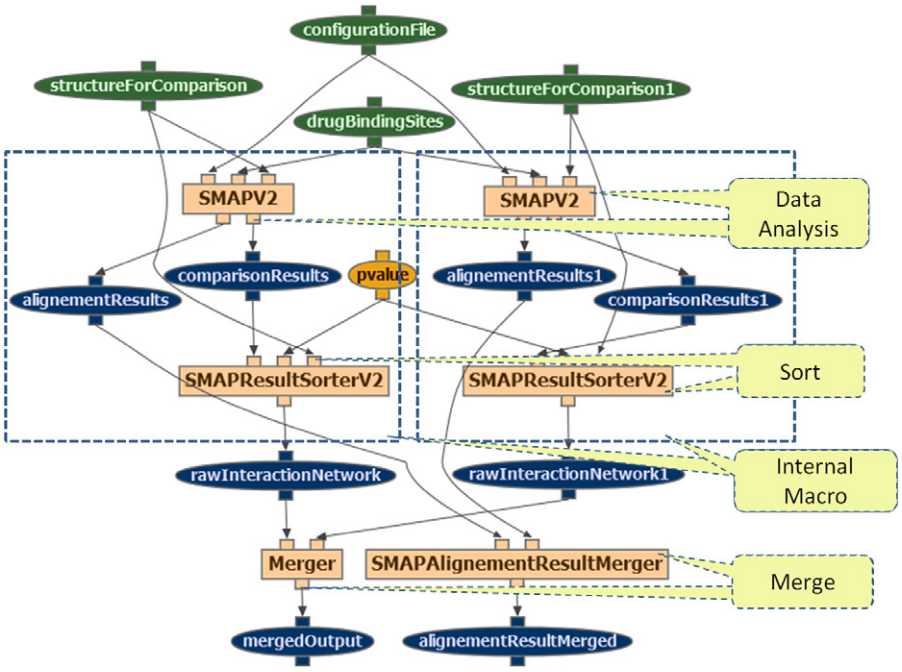


Fig. 1. Sample motifs in a Wings workflow fragment for drug discovery. A comparison analysis is performed on two different input datasets (SMAPV2). The results are then sorted (SMAPResultSorter) and finally merged (Merger, SMAPAlignementResultMerger).

4.1. Data-Operation motifs

4.1.1. Data preparation

Data, once it is retrieved, may need several transformations before being able to be used in a workflow step. The most common activities that we have detected in our analysis are:

4.1.1.1. Combine. Data merging or joining steps are commonly found across workflows. The Combine motif refers to the step or group of steps in the workflow aggregating information from different inputs. An example can be seen in Fig. 1, where the results of both branches of a workflow fragment used for drug discovery are merged for presenting a single output result.

4.1.1.2. Filter. The datasets brought into a pipeline may not be subject to analysis in their entirety. Data could further be filtered, sampled or could be subject to extraction of various subsets.

4.1.1.3. Format transformation. Heterogeneity of formats in data representation is a known issue in many scientific disciplines. Workflows that bring together multiple access or analysis activities usually contain steps for format transformations, sometimes called "Shims" [31], that typically preserve the contents of data while converting its representation format.

4.1.1.4. Input augmentation. Data access and analysis steps that are handled by external services or tools typically require well formed query strings or structured requests as input parameters. Certain tasks in workflows are dedicated to the generation of these queries through an aggregation of multiple parameters. An example of this is provided in the workflow of Fig. 2: For each service invocation (e.g. *getJobState*) there are steps (e.g. *getJobState_Input*) that are responsible for creating the correctly formatted inputs for the service.

4.1.1.5. Output extraction. Outputs of data access or analysis steps could be subject to data extraction to allow the conversion of data from the service format to the workflow internal data carrying structures. This motif is associated with steps that perform the inverse operation of Input Augmentation. An example is given in Fig. 2, where output splitting steps (e.g. *getJobState_output*) are responsible for parsing the result XML message returned from the service (*getJobState*) to return a singleton value containing solely the job state.

4.1.1.6. Group. Some steps of the workflow reorganize the input into different groups or aggregate the inputs on a given collection of data items. For example, grouping a table by a certain category or executing a SQL GROUP-BY clause on an input dataset.

4.1.1.7. Sort. In certain cases datasets containing multiple data items/records are to be sorted (with respect to a parameter) prior to further processing. The Sort motif refers to those steps. Fig. 1 shows an example where the inputs resulting from the data analysis (*comparisonResults*) are sorted (*ComparisonResultsV2* component) before being merged in a subsequent step.

4.1.1.8. Split. Our analysis has shown that many steps in the workflows separate an input (or group of inputs) into different outputs. For example, the splitting of a dataset in different subsets to be processed in parallel in a workflow.

4.1.2. Data analysis

This motif refers to a rather broad category of tasks in diverse domains, and it is highly relevant because it often represents the main functionality of the workflow. An important number of workflows are designed with the purpose of analyzing or evaluating different features of input data, ranging from simple comparisons between the datasets to complex protein analysis to see whether two molecules can be docked successfully or not. An example is given in the workflow of Fig. 2 with a processing step named *warp2D*, and the steps named *SMAPV2* in Fig. 1 with a ligand binding sites comparison of the inputs.

4.1.3. Data cleaning/curation

We have observed the steps for cleaning and curating data as a separate category from data preparation and filtering. Typically these steps are undertaken by sophisticated tooling/services, or by

Please cite this article in press as: D. Garijo, et al., Common motifs in scientific workflows: An empirical analysis, Future Generation Computer Systems (2013), http://dx.doi.org/10.1016/j.future.2013.09.018

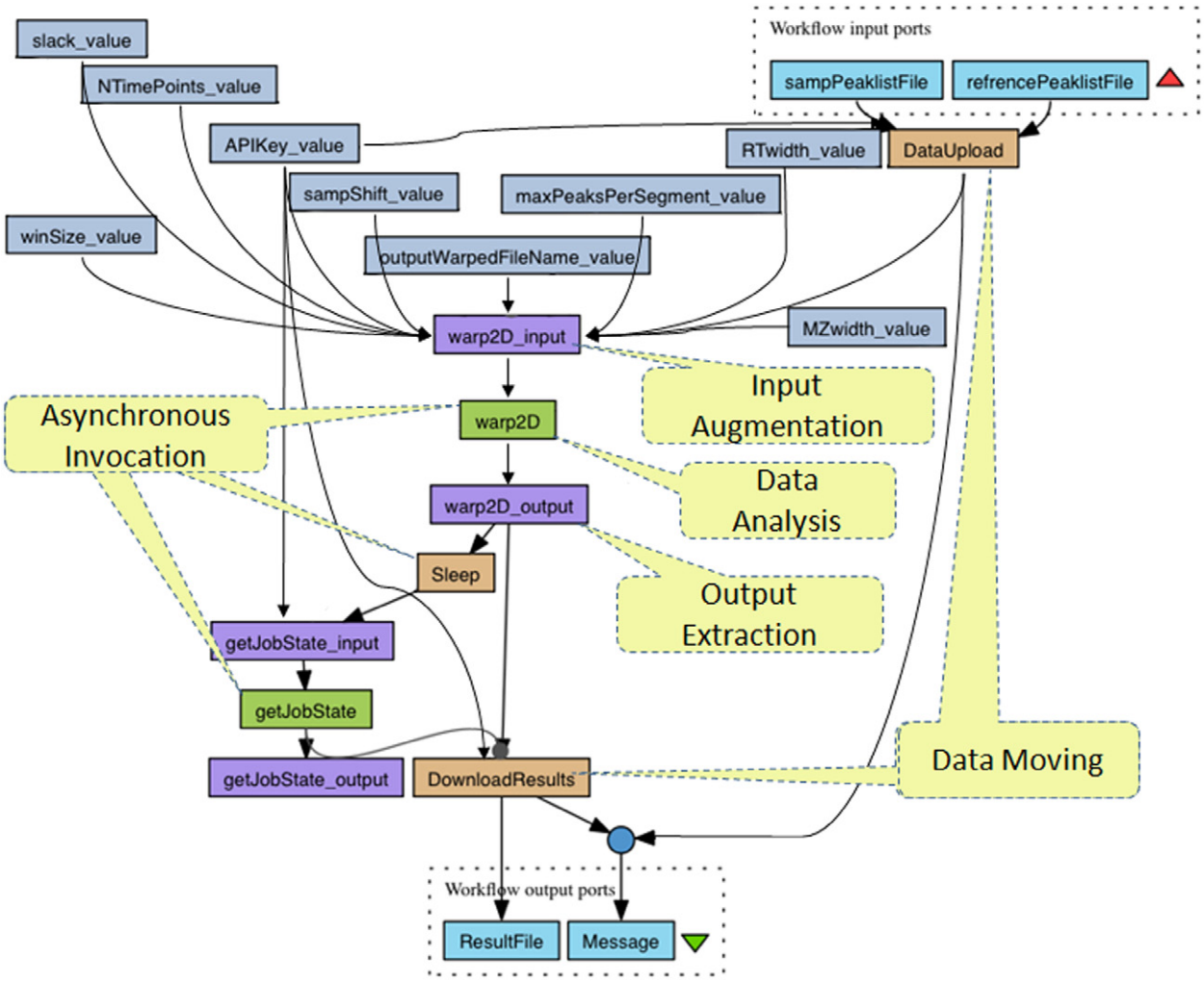


Fig. 2. Sample motifs in a Taverna workflow for functional genomics. The workflow transfers data files containing proteomics data to a remote server and augments several parameters for the invocation request. Then the workflow waits for job completion and inquires about the state of the submitted warping job. Once the inquiry call is returned the results are downloaded from the remote server.

specialized users. A cleaning/curation step essentially preserves and enriches the content of data (e.g., by a user's annotation of a result with additional information, detecting and removing inconsistencies on the data, etc.).

4.1.4. Data movement

Certain analysis activities that are performed via external tools or services require the submission of data to a location accessible by the service/tool (i.e., a web or a local directory respectively). In such cases the workflow contains dedicated step(s) for the upload/transfer of data to these locations. The same applies to the outputs, in which case a data download step is used to chain the data to the next steps of the workflow. The data deposition of the workflow results to a specific server would also be included in this category. In Fig. 2, the *DataUpload* and *DownloadResults* steps ship data to the server on which the analysis will be done, and also retrieve back the results via a dedicated download step.

4.1.5. Data retrieval

Workflows exploit heterogeneous data sources, remote databases, repositories or other web resources exposed via SOAP or REST services. Scientific data deposited in these repositories are retrieved through query and retrieval steps inside workflows. Certain tasks within the workflow are responsible for retrieving data from such external sources into the workflow environment. We also observed that certain data integration workflows contain multiple linked retrieval steps, being essentially parameterized data integration chains.

4.1.6. Data visualization

Being able to show the results is as important as producing them in some workflows. Scientists use visualizations to show the conclusions of their experiments and to take important decisions in the pipeline itself. Therefore, certain steps in workflows are dedicated to generation of plots, graphs, tables, XMLs or Microsoft Excel files outputs from input data. This category is also known as the result delivery of the experimental results.

4.2. Workflow-Oriented motifs

We divide this category in two different groups, depending on whether motifs are observed *among* workflows, by analyzing the relations of the workflow with other workflows (*Inter Workflow Motifs*); or *within* workflows, by exploring the workflow itself (*Intra Workflow Motifs*).

4.2.1. Inter workflow motifs

4.2.1.1. Atomic workflows. Our review has shown that a significant number of workflows perform an atomic unit of functionality,

Please cite this article in press as: D. Garijo, et al., Common motifs in scientific workflows: An empirical analysis, Future Generation Computer Systems (2013), http://dx.doi.org/10.1016/j.future.2013.09.018

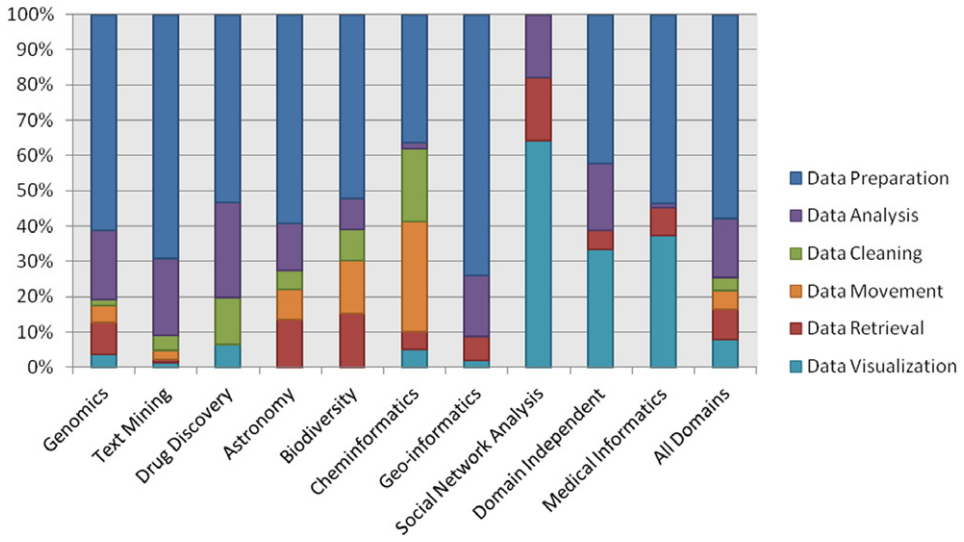


Fig. 3. Distribution of Data-Operation motifs per domain.

which effectively requires no sub-workflow usage. Typically these workflows are designed to be included in other workflows. Atomic workflows are the main mechanism of modularizing functionality within scientific workflows.

4.2.1.2. Composite workflows. The usage of sub-workflows appears as a motif for exploiting modular functionality from multiple workflows. This motif refers to all those workflows that have one or more sub-workflows included in them (in some cases, sub-workflows offer different views of the global workflow, as they could have overlapping steps).

4.2.1.3. Workflow overloading. Our analysis has shown that authors tend to deliver multiple workflows with the same functionality, but operating over different input parameter types. An example is performing an analysis over a String input parameter versus performing it over the contents of a specified file, generalizing a workflow to work with collections of files instead of single inputs, etc. Overloading is a response to the heterogeneity of environments, directly related to workflow reuse (as most of the functionality of the steps in the overloaded workflow remains the same).

4.2.2. Intra workflow motifs

4.2.2.1. Internal macros. This category refers to those groups of steps in the workflow that correspond to repetitive patterns of combining tasks. An example can be seen in Fig. 1, where there are two branches of the workflow performing the same operations in the same order (SMAPV2 plus SMAPResultSorterV2 steps).

4.2.2.2. Human interactions. We have observed that some scientific workflows systems increasingly make use of human interactions to undertake certain activities within workflows. These steps are often necessary to achieve some functionality of the workflow that cannot be (fully) automated, and requires human computation to complete. Typical examples of such activities are manual data curation and cleaning steps (e.g., annotating a Microsoft Excel file), manual filtering activities (e.g., selecting a specific data subset to continue the experiment), etc.

4.2.2.3. Stateful/asynchronous invocations. Certain activities such as analysis or visualizations could be performed through interaction with stateful (web) services that allow for creation of jobs over remote grid environments. Such activities require invocation of multiple operations at a service endpoint using a shared state identifier (e.g. Job ID). An example is given in the workflow of Fig. 2, where the service invocation *warp2D* causes the creation of a stateful warping job. The call then returns a JobID, which is used to inquire about the job status (*getJobStatus*), and to retrieve the results (*DownloadResults*).

5. Workflow analysis results

In this section, we report on the frequencies of the Data-Operation and Workflow-Oriented motifs within the workflows selected for our analysis, and discuss how they are correlated with the workflow domains.[17] A detailed explanation of the frequencies for each domain and for each workflow system can be seen in the Appendix.

Fig. 3 illustrates the distribution of Data-Operation motifs across the domains. The analysis of this figure shows the predominance of the data preparation motif, which constitutes more than 50% of the Data-Operation motifs in the majority of domains. This is an interesting result as it implies that data preparation steps are more common than any other activity, specifically those that perform the main (scientifically-significant) functionality of the workflow. The abundance of these steps is one major obstacle for understandability, since they burden the documentation function and create difficulties for the workflow reuser scientists. The Social Network Analysis domain is an exception, as it consults, analyzes and visualizes queries and statistics over concrete data sources without performing any data preparation steps. Fig. 3 also demonstrates that within domains such as Genomics, Astronomy, Medical Informatics or Biodiversity, where curated common scientific databases exist, workflows are used as data retrieval clients against these databases.

Drilling down to Data Preparation, Fig. 4 shows the dominance of Filter, Input Augmentation and Output Extraction motifs for most domains. Input Augmentation and Output Extraction are activities which can be seen as adapters that help plugging data analysis capabilities into workflows. Their number is higher in workflows relying on third party services, i.e., most Taverna domains (Biodiversity, Cheminformatics, Geo-Informatics); while Filtering is higher in Wings, Galaxy and Vistrails workflows. Fig. 4 also demonstrates how the existence of a widely used common data

[17] Results available at http://www.myexperiment.org/packs/364.html.

Please cite this article in press as: D. Garijo, et al., Common motifs in scientific workflows: An empirical analysis, Future Generation Computer Systems (2013), http://dx.doi.org/10.1016/j.future.2013.09.018

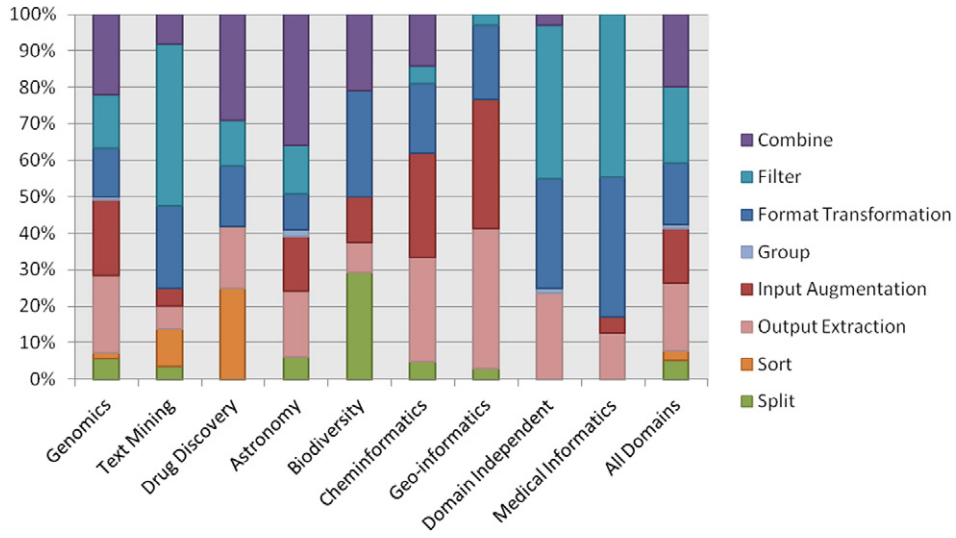


Fig. 4. Distribution of data preparation motifs per domain. The Social Network Analysis domain is not included, as it does not have any data preparation motifs.

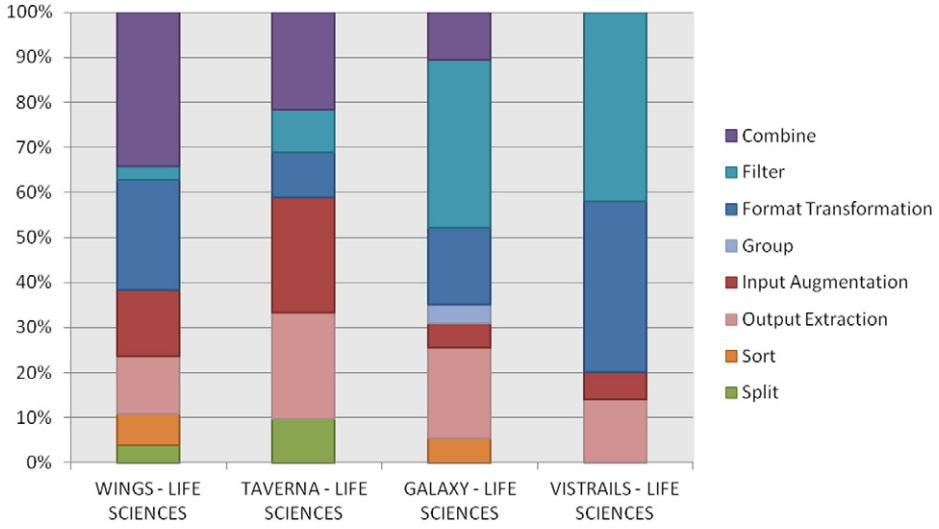


Fig. 5. Data preparation motifs in the Life Sciences workflows.

structure for a domain, in this case the VOTable in Astronomy.[18] reduces the need for domain-specific data transformations in workflows.

Some of the differences between the systems are reflected in the motifs results. As displayed in the comparative Fig. 5 for the Life Sciences domain (a general domain shown in Table 5 including the Genomics, Drug discovery, Biodiversity, Chemical Informatics and Medical Informatics domains), in Wings, Galaxy and Vistrails input augmentation and output extraction steps are much less required (around 30%, 20% and 20% respectively versus almost 50% in Taverna) as the inputs are either controlled (Galaxy, Vistrails) or strongly typed (Wings) and the data analysis steps are pre-designed to operate over specific types of data. Within Fig. 6 we observe that Wings workflows do not contain any data retrieval or movement steps, as data is pre-integrated into the workflow environment (data shipping activities are carried out behind the scenes by the Wings execution environment) whereas in Taverna the workflow carries dedicated steps for querying databases and shipping data to necessary locations for analysis. Galaxy and Vistrails also include components to retrieve content from external

Table 5
Distribution of workflows from Taverna (T), Wings (W), Galaxy (G) and Vistrails (V) in the Life Sciences domain.

Domain	No. of workflows	Source
Drug discovery	7	W
Biodiversity	12	T
Cheminformatics	7	T
Genomics	90	T (38), W (28), G (23), V (1)
Medical informatics	7	V
Total	123	

datasets into the environment (2% and 10% respectively), although we did not find steps for moving the data of intermediate steps to external services among the set of workflows analyzed. In the case of Galaxy this happens because most data retrieval and moving steps are performed transparently to the workflow execution (individual components are used to retrieve the data to the user domain, and that data is then used as input of the workflow); while in Vistrails the main analysis steps of the analyzed workflow set were performed using custom components.

Another interesting finding is the amount of visualization steps found in the Life Science domain (Fig. 6). One feature of Vistrails and Galaxy is the tools included for the visualization of

[18] http://www.ivoa.net/Documents/VOTable/.

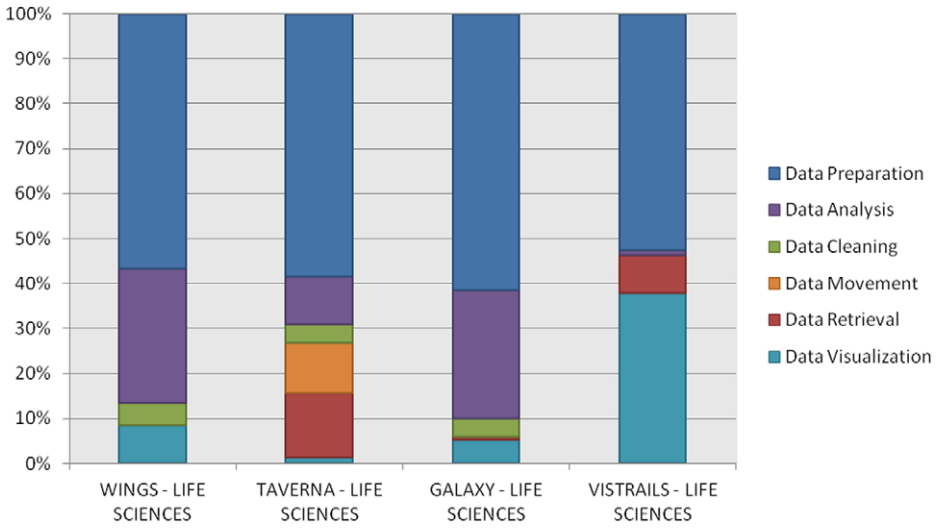


Fig. 6. Data-Operation motifs in the Life Sciences workflows.

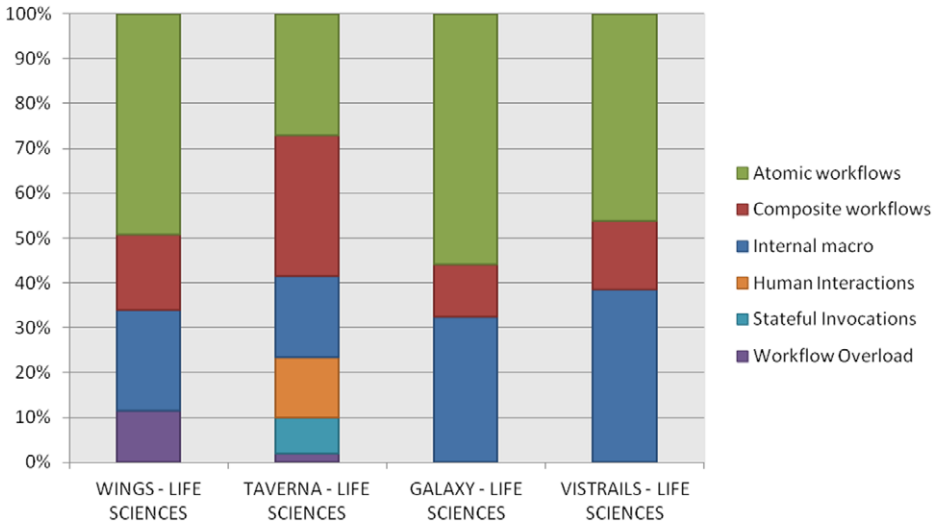


Fig. 7. Workflow-Oriented motifs in the Life Sciences workflows.

workflow results. In Vistrails workflows almost 40% of the motifs found are visualization steps, but this percentage is very reduced in Galaxy (less than 5%). This is due to a separate visualization tool in Galaxy[19] which reduces the need for visualization steps in the workflows. As shown in Fig. 6, the visualization steps in Taverna and Wings are considerably smaller (around 2% and 10% respectively).

The impact of the difference in the execution environments of the analyzed workflow systems is also observed on the Workflow-Oriented motifs, as can be seen in Fig. 7. Stateful invocations motifs are not present in Wings, Galaxy and Vistrails workflows, as all steps are handled by a dedicated workflow scheduling framework/pipeline system and the details are hidden from the workflow developers. In Taverna's default configuration, there are no execution environments or scheduling frameworks prescribed to the users. Therefore the workflow developers are (1) either responsible for catering for various different invocation requirements of external resources, which may include stateful invocations requiring execution of multiple consecutive steps in order to undertake a single function (2) they can develop specific plug-ins that wrap-up stateful interactions and boiler plate steps.

Regarding Workflow-Oriented motifs, Fig. 8 shows that human interaction steps are increasingly used in scientific workflows, especially in the Biodiversity and Cheminformatics domains. Human interaction steps in Taverna workflows are handled either through external tools (e.g., Google Refine), facilitated via a human-interaction plug-in, or through local scripts (e.g., selection of configuration values from multi-choice lists). However, we observed that several boiler-plate *set-up* steps are required for the deployment and configuration of external tooling to support the human tasks. These boiler plate steps result in very large and complex workflows. Wings and Vistrails workflows do not contain human interaction steps. Galaxy is an environment that is heavily based on user-driven configuration and invocation of analysis tools (some parameters and inputs of the workflows can even be changed after the execution of the workflow has started). However, based on our definition of Human Interactions, i.e. analytical data processing undertaken by a human, the Galaxy workflows that we have analyzed do not contain any human computation steps either.

Fig. 8 also shows a large proportion of the combination of Composite Workflows and Atomic Workflows motifs (up to more than 60%) which confirm that the use of sub-workflows is an

19 https://main.g2.bx.psu.edu/visualization/trackster.

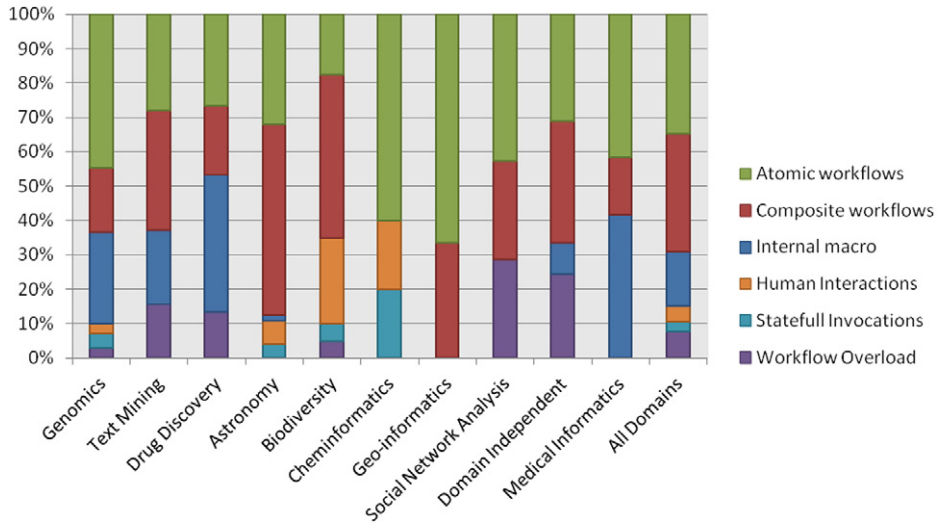


Fig. 8. Distribution of Workflow-Oriented motifs in the analyzed workflows per domain.

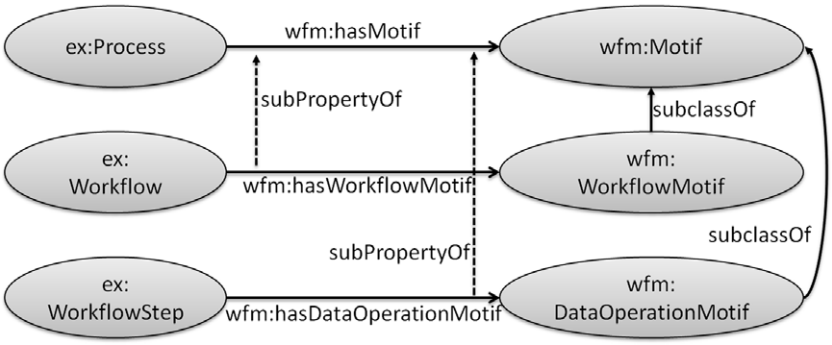


Fig. 9. Diagram showing an overview of the structure of the Motif Ontology. The `wfm` prefix is used for describing the different classes and properties of the Motif Ontology, while the ex prefix is used as a placeholder for any vocabulary for describing workflows and their steps.

established best practice for modularizing functionality and reuse. Sub-workflows can also be used to encapsulate the preparation steps and multi-step service invocations within "Components" [32], in order to reduce obfuscation. These components have well-defined interfaces, support for complex data typing, interface metadata and built-in error-handling.[20] The Workflow Overload motif also plays a relevant role, appearing in almost 10% of the analyzed workflows. Workflows containing this motif are considered advanced forms of sub-workflow development. By executing workflows in different settings, the authors provide overloaded versions of the same functionality in different workflows to increase the coverage on target uses. While we observe overloading as a good practice, a significant behavior of workflow developers in the Taverna environment is to extend their workflows with the ability to accept input from multiple ports in different formats. We believe that such overloading behavior within a single workflow is a poor practice and should be avoided. Instead, multiple workflows operating a single designated input format should be provided.

6. Motif Ontology and annotation of scientific workflows

Our ultimate objective is to provide a catalog of motifs. We expect that this catalog will be used to annotate workflows to denote the nature of activities occurring in them. These annotations would allow (1) helping the creators to describe the particular functionality of the workflows to reach a broader audience of possible reusers, (2) helping in the creation of new workflows by assisting users (e.g., suggesting components based in our motif catalog); and (3) helping in the search of workflows with certain functionality (e.g., workflows with data retrieval, analysis and filtering). This would also be beneficial from a workflow designer perspective, so as to obtain workflows that are similar to the ones being designed.

6.1. Representing motifs

In order to provide workflow designers and curators with the means to annotate, we have designed an OWL ontology[21] that captures the motifs detected in our empirical analysis. Fig. 9 illustrates the basic structure of the ontology. The right hand side of the figure shows how the motifs are organized: the class `wfm:Motif` represents the different classes of motifs identified in Section 4 (`wfm` stands for the prefix the Motif Ontology, with namespace URI http://purl.org/net/wf-motifs). This class is categorized into two specialized sub-classes `wfm:DataOperationMotif` and `wfm:WorkflowMotif`, which are sub-classed according to the taxonomy represented in Table 4.

The ontology provides the `wfm:hasMotif` property in order to associate workflows and their operations with their motifs. The properties `wfm:hasDataOperationMotif` and `wfm:`

[20] http://heater.cs.man.ac.uk:4040/web-wf-design-neiss/index.html.

[21] http://purl.org/net/wf-motifs.

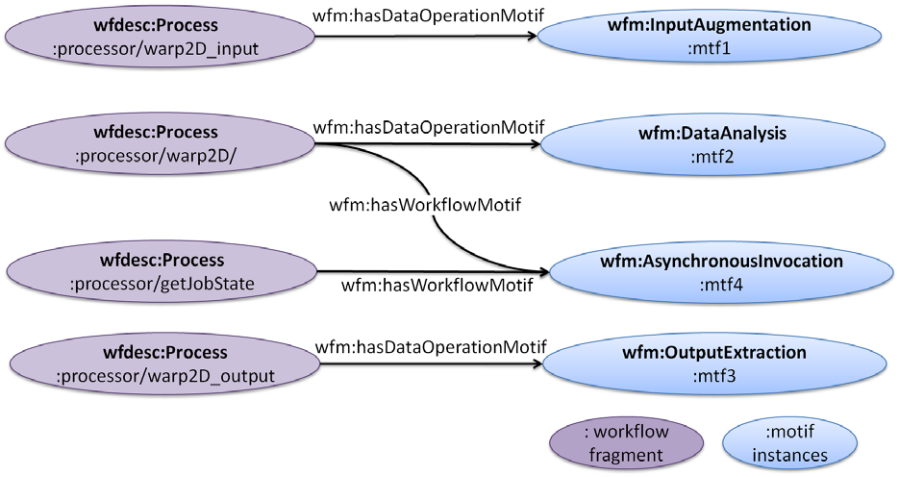


Fig. 10. Subset of the annotations of the Taverna workflow shown in Fig. 2 using the Wfdesc model.

`hasWorkflowMotif` allow annotating workflows and their steps with more specificity. These properties have no domain specified, as different workflow models may use different vocabularies for describing workflows and their parts. Therefore, in Fig. 9 workflows, steps and processes are used as a place holder using the `ex` prefix.

6.2. Representing workflows and workflow steps

Workflows may be represented with different models and vocabularies like Wfdesc [33], OPMW [30], P-Plan [34] or D-PROV [35]. While providing an abstract and consistent representation of the workflow is not a pre-requisite to the usage of the Motif Ontology, we consider it a best-practice to use a model that is independent from any specific workflow language or technology. An example of annotation using the Wfdesc model is given in Fig. 10 by exposing the annotations of part of the Taverna workflow shown Fig. 2.

The annotations encoded using the Motif Ontology could be used for several purposes. By providing explicit semantics on the data processing characteristics and the implementation characteristics of the operations, annotations improve understandability and interpretation. Moreover, they can be used to facilitate workflow discovery. For example, the user can issue a query to identify workflows that implement a specific flow of data manipulation and transformation (e.g., *return the workflows in which data reformatting is followed by data filtering and then data visualization*). Furthermore, having information on characteristics of workflow operations allows for manipulation of workflows to generate summaries. As described in [36], motif markup allows users to specify workflow reduction rules based on motifs (e.g. eliminate data preparation, organization steps, group all steps that belong to the same asynchronous invocation, etc.).

7. Conclusions

The difficulty in understanding the function of workflows is an impediment to reusing and re-purposing scientific workflows. To address this problem, motifs that provide high level descriptions of the tasks carried out by workflow steps can be effective. As a step towards this goal, we reported in this paper on an empirical analysis[22] that we conducted using Taverna, Wings, Galaxy and Vistrails workflows with the objective of identifying the motifs embedded within those workflows. In doing so, *we have defined a catalog of motifs* distinguishing *Data-Operation motifs*, which describe the tasks carried out by the workflow steps, from *Workflow-Oriented motifs*, which describe the way those tasks are implemented within the workflow. It is worth mentioning that, although important, motifs that have to do with scheduling and distributing the execution of workflows [37] or the control flow of the workflow [13] have been left out of the scope of this paper.

We identified 6 major types of Data-Operation motifs and 6 types of Workflow-Oriented motifs that are used across different domains. We *created a Motif Ontology based on the motif catalog* that provides users with the necessary vocabulary to annotate workflows with high-level motifs to facilitate understanding. Part of our current work is the annotation of the motifs in workflows belonging to a provenance repository [38], in order to improve the existent workflow descriptions.

The frequency by which the motifs appear depends on the *differences among the workflow environments* and *differences in domains*. Regarding data preparation motifs (the most common type of motifs), we found that their use is correlated with the type of execution environment for which the workflow is designed. In particular, in a workflow system such as Taverna, which by default does not require data and tools to be embedded into the execution environment, many steps in the workflow can be dedicated to the moving and retrieval of heterogeneous datasets, and stateful resource access protocols. On the other hand, in a workflow system such as Wings, Vistrails and Galaxy we notice that some data preparation motifs, such as data moving, are minimal and in certain domains absent. This happens either because data is pre-integrated into the workflow execution environment or because data primarily exists in external environments and the workflow execution engine performs these operations transparently to the user. The differences among the systems highlight that specialized resource or data access components/plug-ins and standardization in data formats would contribute significantly to the simplification of scientific workflows.

The workflows used in our analysis are taken from a variety of heterogeneous domains and have been crafted by a group of different domain experts scientists. The distribution of the cohort studied among the domains is not even, as their number, availability and documentation differs for each workflow system. Our catalog of motifs refers to the workflows included in the analysis (with special relevance of the Life Science domain), but our intuition is that *most of the motifs will be found in other domains and in other workflow systems*. Future work expanding the analysis

[22] Contents available at http://www.oeg-upm.net/files/dgarijo/motifAnalysisSite/.

Table A.1
Distribution of the data preparation motifs among the workflows analyzed, grouped by domain.

	Combine	Filter	Format transformation	Group	Input augmentation	Output extraction	Sort	Split	Data preparation
Genomics	93	62	56	4	87	90	6	24	422
TextMining	12	64	33	0	7	9	15	5	145
Drug discovery	7	3	4	0	0	4	6	0	24
Astronomy	77	29	21	4	32	39	0	13	215
Biodiversity	5	0	7	0	3	2	0	7	24
Cheminformatics	3	1	4	0	6	6	0	1	21
Geo-informatics	0	1	7	0	12	13	0	1	34
Social network analysis	0	0	0	0	0	0	0	0	0
Domain independent	2	27	19	1	0	15	0	0	64
Medical informatics	0	21	18	0	2	6	0	0	47

Table A.2
Distribution of the Data-Operation motifs among the workflows analyzed, grouped by domain.

	Data preparation	Data analysis	Data cleaning	Data movement	Data retrieval	Data visualization	Total data operation
Genomics	422	134	13	32	63	26	690
TextMining	145	46	9	5	2	3	210
Drug discovery	24	12	6	0	0	3	45
Astronomy	215	48	20	30	48	2	362
Biodiversity	24	4	4	7	7	0	46
Cheminformatics	21	1	12	18	3	3	58
Geo-informatics	34	8	0	0	3	1	46
Social network analysis	0	5	0	0	5	18	28
Domain independent	64	29	0	0	8	51	152
Medical informatics	47	1	0	0	7	33	88

Table A.3
Distribution of the Workflow-Oriented motifs among the workflows analyzed, grouped by domain.

	Atomic workflow	Composite workflow	Internal macro	Human interaction	Stateful invocation	Workflow overload	Total workflow motifs
Genomics	62	26	37	4	6	4	139
Text Mining	25	31	19	0	0	14	89
Drug discovery	4	3	6	0	0	2	15
Astronomy	33	57	2	7	4	0	103
Biodiversity	7	19	0	10	2	2	40
Cheminformatics	3	0	0	1	1	0	5
Geo-informatics	4	2	0	0	0	0	6
Social network analysis	3	2	0	0	0	2	7
Domain independent	14	16	4	0	0	11	45
Medical informatics	5	2	5	0	0	0	12

on other systems like Kepler, Pegasus or ASKALON will be required to further validate our findings.

As part of our future work, we envisage *deriving best practices* that can be used in workflow design and *providing tools that assist users in automatic workflow annotation* using our Motif Ontology. In particular, in an environment like Wings, where semantic typing is supported, it could be possible to automatically detect some Data Preparation activities by inferencing over the types of inputs and the outputs and the task types. In an open environment like Taverna, such classifications are not available, but there are other resources for inferring functionality of steps, like controlled tags on services and the names of processors. In a controlled environment like Galaxy, the modules are already classified in a taxonomy, which could be used to infer some of the proposed motifs. Vistrails, on the other hand, normally produces well documented modules in their workflows. Such documentation could be used to derive the motifs of the workflow. Our identification of Workflow-Oriented motifs also acts as a set of heuristics for automatically creating abstractions over workflows [39], like grouping stateful interactions on a service endpoint, detection of data preparation activities to highlight the real functionality of the workflow, detecting subgroups of repeated data preparations steps (i.e., internal macros), etc.

Finally, another area for future work is to *analyze how motifs could be used to manage workflow decay*. By understanding the functionality of the workflow steps, it should be easier to replace broken steps or fragments with alternative or updated services. In order to achieve this goal, a further exploration of our motifs in combination with the intrinsic and environmental characteristics of the workflows [17] will be required.

Acknowledgments

This research was supported by the Wf4Ever European project (FP7-270192), a grant from the US Air Force Office of Scientific Research (AFOSR) through award number FA9550-11-1-0104, the FPU grant (Formación de Profesorado Universitario) from the Spanish Ministry of Science and Innovation (MCINN) and the myGrid platform grant (EPSRC EP/G026238/1 myGrid: A platform for e-Biology Renewal). The authors would like to thank the many scientists and workflow developers that created the workflows used in this study and generously made them available to the community. We also would like to thank Pinar Karagoz for her comments on possible ways of inferring motifs over existing workflow scripts and Bamdad Dashtban for providing useful references for our work.

Appendix

This section details the occurrences of the motifs for all the workflows measured in the analysis. Tables A.1–A.3 provide details

Table A.4
Distribution of the data preparation motifs among the workflows analyzed, grouped by workflow system.

	Combine	Filter	Format transformation	Group	Input augmentation	Output extraction	Sort	Split	Data preparation
Wings	42	67	50	0	15	16	21	8	219
Taverna	143	60	64	4	126	127	0	43	567
Galaxy	12	48	17	4	5	19	6	0	111
Vistrails	2	33	38	1	3	22	0	0	99

Table A.5
Distribution of the Data-Operation motifs among the workflows analyzed, grouped by workflow system.

	Data preparation	Data analysis	Data cleaning	Data movement	Data retrieval	Data visualization	Total data operation
Wings	219	124	9	0	5	36	393
Taverna	567	112	49	92	124	9	953
Galaxy	111	46	6	0	1	8	172
Vistrails	99	6	0	0	16	87	208

Table A.6
Distribution of the Workflow-Oriented motifs among the workflows analyzed, grouped by workflow system.

	Atomic workflow	Composite workflow	Internal macro	Human interaction	Stateful invocation	Workflow overload	Total workflow motifs
Wings	48	41	31	0	0	27	147
Taverna	75	108	22	22	13	8	248
Galaxy	22	4	11	0	0	0	37
Vistrails	15	5	9	0	0	0	29

of the number of motifs per workflows grouped by domain, while Tables A.4–A.6 specify the occurrences of each motif grouped by the workflow system.

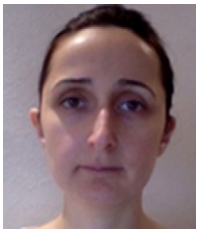
References

[1] Ewa Deelman, Dennis Gannon, Matthew Shields, Ian Taylor, Workflows and e-science: an overview of workflow system features and capabilities, Future Generation Computer Systems 25 (5) (2009) 528–540.
[2] Katherine Wolstencroft, Robert Haines, Donal Fellows, Alan Williams, David Withers, Stuart Owen, Stian Soiland-Reyes, Ian Dunlop, Aleksandra Nenadic, Paul Fisher, Jiten Bhagat, Khalid Belhajjame, Finn Bacall, Alex Hardisty, Abraham Nieva de la Hidalga, Maria P. Balcazar Vargas, Shoaib Sufi, Carole Goble, The taverna workflow suite: designing and executing workflows of web services on the desktop, web or in the cloud, Nucleic Acids Research (2013).
[3] Yolanda Gil, Varun Ratnakar, Jihie Kim, Pedro A. González-Calero, Paul T. Groth, Joshua Moody, Ewa Deelman, Wings: intelligent workflow-based design of computational experiments, IEEE Intelligent Systems 26 (1) (2011) 62–72.
[4] Jeremy Goecks, Anton Nekrutenko, James Taylor, Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences, Genome Biology 11 (8) (2010) R86.
[5] Carlos E. Scheidegger, Huy T. Vo, David Koop, Juliana Freire, Claudio T. Silva, Querying and re-using workflows with vistrails, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD'08, ACM, New York, NY, USA, 2008, pp. 1251–1254.
[6] Bertram Ludäscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A. Lee, Jing Tao, Yang Zhao, Scientific workflow management and the kepler system, Concurrency and Computation: Practice and Experience 18 (10) (2006) 1039–1065.
[7] Thomas Fahringer, Radu Prodan, Rubing Duan, Jürgen Hofer, Farrukh Nadeem, Francesco Nerieri, Jun Qin Stefan Podlipnig, Mumtaz Siddiqui, Hong-Linh Truong, Alex Villazón, Marek Wieczorek, ASKALON: a development and grid computing environment for scientific workflows, in: Ian J. Taylor, Ewa Deelman, Dennis B. Gannon, Matthew Shields (Eds.), Workflows for e-Science, Springer, 2007, pp. 450–471 (Chapter 27).
[8] Jia Zhang, Wei Tan, J. Alexander, I. Foster, R. Madduri, Recommend-as-you-go: a novel approach supporting services-oriented scientific workflow reuse. in: Proceeding of IEEE International Conference on Services Computing, SCC, Washington, DC, July 2011, pp. 48–55.
[9] Antoon Goderis, Ulrike Sattler, Phillip W. Lord, Carole A. Goble, Seven bottlenecks to workflow reuse and repurposing, in: International Semantic Web Conference, Springer, 2005, pp. 323–337.
[10] David De Roure, Carole A. Goble, Robert Stevens, The design and realisation of the myexperiment virtual research environment for social sharing of workflows, Future Generation Computer Systems 25 (5) (2009) 561–567.
[11] Phillip Mates, Emanuele Santos, Juliana Freire, Cláudio T. Silva, Crowdlabs: social analysis and visualization for the sciences, in: 23rd International Conference on Scientific and Statistical Database Management, SSDBM, Springer, 2011, pp. 555–564.
[12] Daniel Garijo, Pinar Alper, Khalid Belhajjame, Oscar Corcho, Yolanda Gil, Carole Goble, Common motifs in scientific workflows: an empirical analysis, in: 8th IEEE International Conference on eScience 2012, Chicago, IEEE Computer Society Press, USA, 2012.
[13] Wil M.P. van der Aalst, Arthur H.M. ter Hofstede, Bartek Kiepuszewski, Alistair P. Barros, Workflow patterns, Distributed and Parallel Databases 14 (1) (2003) 5–51.
[14] Sara Migliorini, Mauro Gambini, Marcello La Rosa, Arthur H.M. ter Hofstede, Pattern-based evaluation of scientific workflow management systems. Technical Report, Queensland University of Technology, 2011, URL: http://eprints.qut.edu.au/39935/.
[15] Malcolm Atkinson, Rob Baxter, Paolo Besana, Michelle Galea, Mark Parsons, Peter Brezany, Oscar Corcho, Jano van Hemert, David Snelling (Eds.), THE DATA BONANZA—Improving Knowledge Discovery for Science, Engineering and Business, WILEY-INTERSCIENCE, A John Wiley & Sons, Inc. Publication, Edinburgh, UK, 2013.
[16] Lavanya Ramakrishnan, Beth Plale, A multi-dimensional classification model for scientific workflow characteristics, in: Proceedings of the 1st International Workshop on Workflow Approaches to New Data-centric Science, Wands'10, ACM, New York, NY, USA, 2010, pp. 4:1–4:12.
[17] Simon Ostermann, Radu Prodan, Thomas Fahringer, Ru Iosup, Dick Epema, On the characteristics of grid workflows, in: Proceedings of CoreGRID Integration Workshop 2008, Hersonisson, Crete, 2008, pp. 431–442.
[18] Sarah Cohen-Boulakia, Jiuqiang Chen, Paolo Missier, Carole Goble, Alan R Williams, Christine Froidevaux, Distilling structure in taverna scientific workflows: a refactoring approach, BMC Bioinformatics (2013) to appear.
[19] R.D. Stevens, C.A. Goble, P. Baker, A. Brass, A classification of tasks in bioinformatics, Bioinformatics 17 (2) (2001) 180–188.
[20] Yolanda Gil, Paul Groth, Varun Ratnakar, Christian Fritz, Expressive reusable workflow templates, in: Proceedings of the Fifth IEEE International Conference on e-Science, Oxford, UK, 2009.
[21] Ingo Wassink, Paul E Van Der Vet, Katy Wolstencroft, Pieter B T Neerincx, Marco Roos, Han Rauwerda, Timo M Breit, Analysing scientific workflows: why workflows not only connect web services, Published in 2009 World Conference on Services - I., 2009, pp. 314–321 (5).
[22] Johannes Starlinger, Sarah Cohen-Boulakia, Ulf Leser, (Re)use in public scientific workflow repositories, in: Anastasia Ailamaki, Shawn Bowers (Eds.), Scientific and Statistical Database Management, in: Lecture Notes in Computer Science, vol. 7338, Springer, Berlin, Heidelberg, 2012, pp. 361–378.
[23] Asunción Gómez Pérez, Richard Benjamins, Applications of ontologies and problem-solving methods, AI Magazine 20 (1) (1999).
[24] Jose Manuel Gómez-Pérez, Michael Erdmann, Mark Greaves, Oscar Corcho, Richard Benjamins, A framework and computer system for knowledge-level acquisition, representation, and reasoning with process knowledge, International Journal of Human–Computer Studies 68 (10) (2010).
[25] E. Deelman, G. Singh, M. Su, J. Blythe, Y. Gil, C. Kesselman, J. Kim, G. Mehta, K. Vahi, G.B. Berriman, J. Good, A. Laity, J.C. Jacob, D.S. Katz, Pegasus: a framework for mapping complex scientific workflows onto distributed systems, Scientific Programming 13 (3) (2005).

Please cite this article in press as: D. Garijo, et al., Common motifs in scientific workflows: An empirical analysis, Future Generation Computer Systems (2013), http://dx.doi.org/10.1016/j.future.2013.09.018

[26] Chris A. Mattmann, Daniel J. Crichton, Nenad Medvidovic, Steve Hughes, A software architecture-based framework for highly distributed and data intensive scientific applications, in: Proceedings of the 28th International Conference on Software Engineering, ICSE'06, ACM, New York, NY, USA, 2006, pp. 721–730.
[27] B. Giardine, et al., Galaxy: a platform for interactive large-scale genome analysis, Genome Research 15 (10) (2005) 1451–1455.
[28] Daniel Blankenberg, James Taylor, Ian Schenck, Jianbin He, Yi Zhang, Matthew Ghent, Narayanan Veeraraghavan, Istvan Albert, Webb Miller, Kateryna D Makova, et al., A framework for collaborative analysis of encode data: making large-scale analyses biologist-friendly, Genome Research 17 (6) (2007) 960–964.
[29] Steven P. Callahan, Juliana Freire, Emanuele Santos, Carlos E. Scheidegger, Cludio T. Silva, Huy T. Vo, Vistrails: visualization meets data management, in: ACM SIGMOD, ACM Press, 2006, pp. 745–747.
[30] Daniel Garijo, Yolanda Gil, A new approach for publishing workflows: abstractions, standards, and linked data, in: Proceedings of the 6th Workshop on Workflows in Support of Large-Scale Science, ACM, Seattle, 2011, pp. 47–56.
[31] Duncan Hull, Robert Stevens, Phillip Lord, Chris Wroe, Carole Goble, Treating shimantic web syndrome with ontologies, in: AKT Workshop on Semantic Web Services, 2004.
[32] Kevin Davies, Democratizing informatics for the long tail scientist, Bio-IT World Magazine (2011).
[33] Khalid Belhajjame, Oscar Corcho, Daniel Garijo, Jun Zhao, Paolo Missier, David Newman, Raul Palma, Sean Bechhofer, Esteban Garcia-Cuesta, Jose-Manuel Gomez-Perez, Graham Klyne, Kevin Page, Marco Roos, Jose Enrique Ruiz, Stian Soiland-Reyes, Lourdes Verdes-Montenegro, David De Roure, Carole Goble, Workflow-centric research objects: first class citizens in scholarly discourse, in: Proceedings of Sepublica2012, 2012, pp. 1–12.
[34] Daniel Garijo, Yolanda Gil, Augmenting prov with plans in p-plan: scientific processes as linked data. in: Second International Workshop on Linked Science: Tackling Big Data (LISC), Held in Conjunction with the International Semantic Web Conference, ISWC, Boston, MA, 2012.
[35] Paolo Missier, Saumen Dey, Khalid Belhajjame, Victor Cuevas, Bertram Ludaescher, D-PROV: extending the PROV provenance model with workflow structure, in: Procs. TAPP'13, Lombard, IL, 2013.
[36] Pinar Alper, Khalid Belhajjame, Carole Goble, Pinar Karagoz, Small is beautiful: Summarizing scientific workflows using semantic annotations, in: Proceedings of the IEEE 2nd International Congress on Big Data, BigData 2013, Santa Clara, CA, USA, June 2013.
[37] Arun Ramakrishnan, Gurmeet Singh, Henan Zhao, et al., Scheduling data-intensive workflows onto storage-constrained distributed resources, in: CCGRID, IEEE Computer Society, 2007, pp. 401–409.
[38] Khalid Belhajjame, Jun Zhao, Daniel Garijo, Aleix Garrido, Stian Soiland-Reyes, Pinar Alper, Oscar Corcho, A workflow prov-corpus based on taverna and wings, in: Proceedings of the Joint EDBT/ICDT 2013 Workshops, EDBT'13, ACM, New York, NY, USA, 2013, pp. 331–332.
[39] Daniel Garijo, Óscar Corcho, Yolanda Gil, Detecting common scientific workflow fragments using templates and execution provenance. in: The Seventh International Conference on Knowledge Capture, K-CAP, Banff, Alberta, Canada, 2013.

Daniel Garijo is a Ph.D. student in the Ontology Engineering Group at the Artificial Intelligence Department of the Computer Science Faculty of Universidad Politécnica de Madrid. His research activities focus on e-Science and the Semantic web, specifically on how to increase the understandability of scientific workflows using provenance, metadata, intermediate results and Linked Data.



Pinar Alper is a Ph.D. student at the School of Computer Science of the University of Manchester. Her research work focuses on abstraction of scientific workflows and distillation of provenance information for data publishing. She currently participates in the EU Wf4EVER project and the myGrid project.

Khalid Belhajjame is a Researcher at the University of Manchester. His general research areas are information and knowledge management, where he has contributed to research proposals in the fields of data integration, knowledge engineering of semantic web services, and scientific workflows. He is an active member of the W3C provenance working group, the DataONE scientific workflow and the Wf4Ever EU project.

Oscar Corcho is an Associate Professor at Departamento de Inteligencia Artificial (Facultad de Informática, Universidad Politécnica de Madrid), and he belongs to the Ontology Engineering Group.

His research activities are focused on Semantic e-Science and Real World Internet, although he also works in the more general areas of Semantic Web and Ontological Engineering. In these areas, he has participated in a number of EU projects (Wf4Ever, PlanetData, SemsorGrid4Env, ADMIRE, OntoGrid, Esperonto, Knowledge Web and OntoWeb), and Spanish R&D projects (CENITS mIO!, España Virtual and Buscamedia, myBigData, GeoBuddies), and has also participated in privately-funded projects like ICPS (International Classification of Patient Safety), funded by the World Health Organisation, and HALO, funded by Vulcan Inc.

Yolanda Gil is Principal Investigator and Project Leader of the Interactive Knowledge Capture research group at USC's Information Sciences Institute (ISI). Her research focuses on intelligent interfaces for knowledge capture, which is a central topic in our projects concerning knowledge-based planning and problem solving, information analysis and assessment of trust, semantic annotation tools, agent and software choreography, and community-wide development of knowledge bases. A recent focus is assisting scientists with large-scale applications throughout the design of workflows and their distributed execution.

Carole Goble is a Full Professor at the University of Manchester School of Computer Science, where she co-leads the Information Management Group. She has worked closely with life scientists for many years and has an international reputation in the Semantic Web, e-Science and Grid communities. Carole is the Director of the myGrid project, a team that produce and use a suite of tools designed to "help e-Scientists get on with science and get on with scientists".

Please cite this article in press as: D. Garijo, et al., Common motifs in scientific workflows: An empirical analysis, Future Generation Computer Systems (2013), http://dx.doi.org/10.1016/j.future.2013.09.018