Discovering Scientific Workflows: The ^{my}Experiment Benchmarks

Antoon Goderis, David De Roure, Carole Goble, Jiten Bhagat, Don Cruickshank,
Paul Fisher, Danius Michaelides, Franck Tanoh

***Abstract*—Automation in science is increasingly marked by the use of workflow technology. The *sharing* of workflows through publication mechanisms or repositories supports the verifiability, reproducibility and extensibility of computational experiments. However, the subsequent *discovery* of workflows remains a challenge, both from a technological and sociological viewpoint. We investigate current practices in workflow sharing, re-use and discovery amongst life scientists chiefly using the Taverna workflow management system. The study draws on two key sources: (i) a survey of researchers drawn from 19 research labs and (ii) an analysis of scientists' behaviour on the ^{my}Experiment social network site, designed to encourage workflow exchange. The results reveal a multi-modal approach to workflow discovery, based on a mix of search on the content of the workflow and its situated context. We go on to develop a benchmark specifically for the evaluation of workflow discovery and to demonstrate it on two example approaches.**

***Index Terms*—Scientific Workflow, Bioinformatics, myExperiment**

I. Introduction

The process of scientific research has a crucial social element: it involves the sharing and publication of protocols and experimental procedures so that results can be reproduced and properly interpreted, and so that others may re-use, repurpose and extend protocols to support the advancement of science.

Scientific processes are increasingly being captured as Scientific Workflows, as workflow tools are adopted to exploit computational services and to deal systematically with the deluge of data generated by new experimental techniques [DG06] [GDE$^+$07]. An example of such a Scientific Workflow Management System is Taverna [OGA$^+$05], which has been widely adopted across a range of disciplines and is particularly popular in the Life Sciences.

Mechanisms for publishing and sharing scientific workflows are beginning to emerge on the Web. For Taverna alone, we found more than 15 repositories, harboring over 500 workflows.

However, it is not enough simply to publish workflows; faced with an increasing number of workflow systems and an increasing number of workflows, scientists now need assistance in discovering them too.

The myExperiment social web site (www.myexperiment.org) has been designed to address exactly this problem. More than a workflow repository, it explicitly sets out to facilitate workflow sharing within scientific communities and to support workflow discovery and re-use.

To achieve discovery an understanding needs to be developed regarding the necessary characteristics and behaviour of a workflow discovery system. By examining how scientists achieve workflow discovery, we can better support effective finding over a growing body of workflows. This paper presents two main contributions:

1) *A study of discovery practice and attitudes.* As an important step towards achieving this understanding, we have worked with scientists to identify the prevalent attitudes to discovery and discovery behaviour. Our user cohort is drawn from bioinformatics, a domain which makes very significant use of workflows. The study draws on two key sources: (i) a survey of 24 researchers drawn from 19 research labs and (ii) an analysis of scientists' behaviour on the myExperiment site. The results reveal a multi-modal approach to workflow discovery, based on a mix of search on the content of the workflow and its situated context.
2) *Benchmarks established for workflow discovery tools.* This empirical analysis provides both quantitative outcomes and valuable insights, constituting an important step towards providing scientists with a new generation of tools to support their research.

The rest of the paper is structured as follows: Section II defines workflows, workflow re-use and discovery, and introduces the differences between search on the content of the workflow versus search using its context. In Section III we present related work in workflow discovery. In Section IV we present the results of our empirical study of workflow discovery and re-use. Section V presents our benchmarks, designed to address the perceived lack of effective discovery tools, while Section VI demonstrates the benchmarks on two example tools. Section VII concludes and considers future work.

II. Defining workflow discovery

We introduce the following definition of **workflow discovery** and extend it to the case of scientific workflows below.

> Workflow discovery is the *process* of retrieving *orchestrations of services* to *satisfy user information need.*

A. Workflow discovery is a process

Workflow discovery is a process that is manual or automated. Manual workflow discovery does not scale well for

A. Goderis, C. Goble, J. Bhagat, P. Fisher and F. Tanoh are with the School of Computer Science, The University of Manchester, UK e-mail: carole.goble@manchester.ac.uk

D. De Roure, D. Cruickshank and D. Michaelides are with University of Southampton, UK

Manuscript received April 30, 2008.

the individual faced with an increasing number of workflows, but its observation potentially reveals problem-solving patterns that are useful to automated techniques. Automated workflow discovery requires electronic input to enable the process, such as textual queries, navigation based on hyperlinks, tag clouds or even known examples of workflows.

B. Satisfy user information need

Our target users are *scientists* looking for existing workflows that support their research. To be able to satisfy them, we need to document their information need and to evaluate how well retrieval techniques fulfill it.

In earlier work [WGG$^+$07], we documented user information need based on several case studies of scientists recycling workflows created by others. We found it useful to draw a distinction between *workflow re-use*, where workflows and workflow fragments created by one scientist might be used as is, and the more sophisticated *workflow repurposing*, where they are used as a starting point by others.

- A user will **re-use** a workflow or workflow fragment that fits their purpose and could be customised with different parameter settings or data inputs to solve their particular scientific problem.
- A user will **repurpose** a workflow or workflow fragment by finding one that is *close enough* to be the basis of a new workflow for a different purpose and making small changes to its *structure* to fit it to its new purpose.

It is important to realise that the difference between supporting workflow re-use and repurposing leads to *different requirements for the discovery process*. Whereas re-use requires finding workflows that are similar to a given user query (“Find a workflow that produces protein sequence.”), repurposing requires finding both similar workflows (“Find a workflow able to replace my faulty workflow fragment.”) and complementary ones (“Find a workflow that extends my current annotation pipeline with a visualisation step.”).

Figure 1 provides an example of repurposing based on two dataflows. It shows the insertion of service **c** from Workflow 2 in between the previously connected services **a** and **b** of Workflow 1. In terms of the underlying bioinformatics, Workflow 1 is extended with the Transeq service, which changes the workflow from a pipeline for measuring similarity of DNA sequences into one that analyses similarity of peptide sequences.

Observe how Workflow 1 provides useful input to locate Workflow 2 in a repository: one can concentrate the search on those service compositions that acccept service **a**’s output and produce service **b**’s input. Finding compatible insertions is one type of discovery that supports the repurposing of dataflows. The other types are the discovery of replacements and the discovery of extensions that append or prepend a workflow.

In Section V we develop a benchmark measuring how life scientists find similar and complementary workflows. Section VI evaluates two example automated approaches on the benchmark by comparing how well they find workflows.

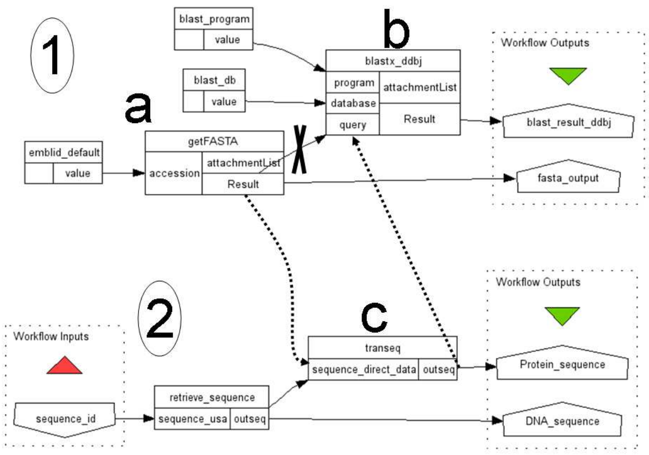


Fig. 1. Example of an insertion based on two Taverna workflows

C. Orchestrations of services

Multiple definitions of a workflow are in use in the scientific community [GDE$^+$07][GL05] and in the business community [GHS95]. What unifies these is the notion that a workflow orchestrates services. What sets them apart are differences in both *workflow content* and *workflow context*.

1) Workflow content: In terms of content, workflows vary on the following dimensions.

a) The kinds of domain and process they represent. For example, this paper draws workflows from the *bioinformatics* domain.
b) The granularity and type of services they orchestrate, e.g. local Beanshell scripts versus a Web Service that provides access to the National Grid Service in the U.K.[1]
c) The workflow language they use, e.g. BPEL[2] or the Simple Conceptual Unified Flow Language (SCUFL) used in the Taverna system. A workflow language relies on one or more models of computation to govern the interaction between services in a workflow [GBA$^+$07]. *Our focus is on models of computation that follow a dataflow paradigm.* Dataflow has proven to be a popular paradigm with scientific workflows that capture transformations on data [GDE$^+$07].
d) The phase of the workflow lifecycle they reflect. The workflow lifecycle entails the following phases:
    i) *During design*, while the workflow is still being designed.
    ii) *Post design, pre-enactment*, as either a finished, concrete workflow where the required resources are known, or as a finished yet abstract workflow (also known as a template) whose resources will be decided dynamically during enactment.
    iii) *During enactment*, when intermediary results come about.
    iv) *Post enactment*, when all results are available.

[1] Web site: www.grid-support.ac.uk
[2] Web site: www.oasis-open.org/committees/wsbpel

Workflow discovery applies to representations capturing any of these phases. *For the purposes of the paper we are interested in finished, concrete workflows.*

2) Workflow context: Also in terms of situational context, there is rich diversity between workflows.

a) Their associated objects. One aspect of context is simply association. Examples include: publications which include workflows; a person who has authored or used workflows; a project that makes workflows available from a Web site; an *in vivo* experiment recorded in a Laboratory Information Management System that launches workflows to find confirmation for hypotheses in Web data. Other associated entities include reviews and discussions on blogs, Wikis and forums.

b) The way they are used. Some workflows are used once, e.g. to compute the Higgs boson at CERN; others repeatedly to monitor new submissions to public databases. Some workflows are popular, others are not.

c) Their annotations. Workflows might be provided in a library or repository where they are explicitly annotated to support discovery. They may have descriptive text, explicit keywords provided by the author, or tags provided by the community. Metadata schema or controlled vocabularies may be in use.

D. Retrieving

As an information retrieval problem, workflow discovery fits inside both a *navigation* and a *search* paradigm [BYRN99]. Both paradigms encompass discovery *by example*. It is clear that workflow discovery is potentially driven from a wide range of descriptions, including existing workflow examples. We distinguish between content-based, context-based and multi-modal discovery.

1) Content-based discovery: Each of the items listed above under workflow content are potential drivers for retrieval. We simply highlight the retrieval of *finished concrete workflows* based on the phases of the workflow lifecycle. In each phase of their lifecycle, workflows are associated with different types of information, all of which yield distinct contexts for the retrieval of concrete workflows.

i) *During design*. The *preliminary design* of a workflow can guide a workflow developer to earlier concrete workflows modelled on a similar design.

ii) *Post design, pre-enactment.* An abstract workflow can serve as input to find a cluster of related concrete workflows. *Elements of a concrete workflow* can serve to find related concrete workflows. A single service can act as a basis to retrieve relevant workflows. Likewise, a selection of a subset of services, void of any control flow, can suffice. Sometimes a workflow fragment or the complete concrete workflow will be relevant, including its control flow. We may also know the provenance of the workflow itself, so that we can ask "Which workflows were derived from this one?" or "Which workflows is this workflow based on?"

iii) *During enactment.* Workflow execution provides an important basis for discovery, and one that distinguishes workflow discovery from discovery of many other digital artefacts. *Partial results of a long-running workflow* can direct a workflow designer to find concrete workflows that can work off these results.

iv) *Post enactment.* When a workflow is executed it uses data and services and it produces data and execution logs. So in principle it would be possible to find candidate workflows by asking "Which workflows have used this data as input?" or "Which workflows have successfully used this service?"

2) Context-based discovery: Workflow can also be discovered based on what they are associated with.

a) Their associated objects. Finding a relevant publication which includes workflows is one example, while another is finding a person who has authored or used workflows. Workflow discovery then derives from searching for publications, people or projects. References to workflows can also occur through reviews and discussion on blogs, wikis and forums.

b) The way they are used. Collaborative filtering techniques very much apply, e.g. recommendations such as "People who used this workflow also used that workflow" or "people who are similar to you have used these workflows".

c) Their annotations. By collecting knowledge of both the wider community and specialised curators, we obtain multiple points of view, which increases chances for serendipitous re-use and discovery.

3) Multi-modal discovery: A mixture of content- and context-based discovery is the third option. We identify two types of mixture:

1) Mixing discovery techniques, where each technique focuses on either workflow content or context. An example would be to look up an author and then search her set of workflows for a particular result.
2) Aggregating content and context into compound *Scientific Research Objects*. myExperiment for instance provides mechanisms for workflows to be collected together with data and other items to form compound *Scientific Research Objects*. This grouping mechanism can be likened to a "shopping cart" or "wish-list" on an Internet shopping site, and enables users to collect items together for various purposes such as giving to others or archiving. Hence we have an additional mechanism of association which can be exploited for workflow discovery.

III. Related work

We relate our work to the current state of the art in discovery within scientific workflow repositories and survey specialised techniques for workflow discovery that are potentially useful for such repositories. We also review existing studies into discovery practice and benchmarking.

A. Discovery support in scientific workflow repositories

We characterise the current situation in finished concrete workflow discovery in Web workflow repositories based on four academic systems (BioWep, INB, Sigenae and Kepler)[3]

[3] Web sites: bioinformatics.istge.it/biowep, www.inab.org/MOWServ, www.sigenae.org/index.php?id=84 and library.kepler-project.org

and two commercial ones (Inforsense and Pipeline Pilot).[4] BioWep, INB and Sigenae offer Taverna workflows; the other systems their own type.

All provide basic discovery capabilities, by searching over workflow titles (Pipeline Pilot) or textual descriptions (BioWep, Sigenae, Kepler, Inforsense). Some systems provide the possibility to search (Kepler, Inforsense) or browse (INB) semantic descriptions. All regard a workflow as an atomic entity, focussing on its overall inputs and outputs, and disregarding its internal structure. None of these systems supports finding workflows based on their context or by example.

B. Techniques for discovery of finished concrete workflows

A number of techniques exist in support of the discovery of finished concrete workflows. Multiple workflow languages are considered, with the scope of each approach being limited to one language. Different data structures represent workflows, with graphs being a popular option, and different techniques work over these structures. Only one of the approaches reports having performed an evaluation with end users.

The Chimera system, [ZWF06] translates workflows available as Virtual Data Language specifications into untyped graphs. The system allows to retrieve workflows by example, in that a query graph can be fed into the system in order to retrieve pipelines extending the one represented by the query graph.

The VisTrails system [SVK$^+$07] enables querying of pipelines of specialised visualization modules from the VTK dataflow-based visualization system. It translates the pipelines into typed graphs and relies on an untyped graph matcher to offer exact (pattern-based) and approximate search. This enables retrieval of pipelines by example. Approximate search is implemented by calculating how one pipeline can be transformed into another and recording the relative ease with which this can be done between all possible pipeline pairs. Finally, it supports discovery of complementary workflows (cfr. the example of Fig. 1).

Relying on the representation of a BPEL workflow as a typed graph, Corrales et al. [CGB06] use error-correcting graph subisomorphism detection. The technique enables them to calculate an edit distance between graphs and hence to define a structural similarity metric for workflows.

Bernstein and Klein [BK02] designed a query language (the Process Query Language) to enable exact structural queries over an Entity Relationship (ER) diagram. The ER diagram in question formalises the structure of processes available in the MIT Process Handbook. Kiefer [KBL$^+$07] translated Process Handbook entries into Resource Description Framework (RDF) graphs and relied on text similarity to retrieve similar graphs.

A Description Logic based approach is explored by Wroe and colleagues [WSG$^+$03]. They abstract Taverna workflows, written in the Scufl language, to be a bag of services and discard all structural relationships between these services. The workflows are represented by concepts in DAML+OIL, a precursor to OWL, the Web Ontology Language.[5] Workflows are represented by concepts that have part-of relationships with other concepts that in turn describe the constituent services of a workflow. Subsumption reasoning is used to detect whether the services in one bag subsume the set of services in another.

Finally, Mahleko and Wombacher [MW06] work over workflows in the form of RosettaNet Partner Interface Processes. Their approach tackles the hard computational complexity the above approaches are typified by, and proposes the use of Finite State Machines (FSMs) to obtain fast performance during matching. The formalisation of a workflow into an FSM can be done with decreasing precision. They evaluated their work in a parallel study on workflow similarity metrics (see below).

C. Studies of discovery practice and benchmarking

There is little work in the *workflow literature* on building human benchmarks. Recent work in the area has aimed to uncover the particular metrics people use for establishing workflow similarity. Bernstein and colleagues [BKBK05] look for the best semantic similarity measures to rank business processes from the MIT Process Handbook, based on a process ontology. The processes are non-executable workflows hence no reuse of workflows in a Web services context is envisioned. The work of Wombacher [Wom06] seeks to elicit the similarity metrics used by workflow researchers when performing the task of comparing the control flow complexity of workflows described by Finite State Machines (FSMs). Data flow is left outside the scope. Wombacher also investigates which metrics, known from workflow mining and FSM techniques, are able to reproduce the human rankings from this task.

In the *service discovery literature*, most of the papers presenting techniques ignore how humans go about discovery and focus instead on a technical evaluation, demonstrating how expressive a technique is, or how scalable. An exception is the work by Dong et al. [DHM$^+$04], who built a small human benchmark based on real Web services to test the performance of the Woogle tool for Web service search. We know of two community initiatives to compare Web service discovery techniques: the Semantic Web Services Challenge and the Web Service Challenge.[6] Both initiatives have limited involvement from users. In the former, a challenging scenario is put forward involving fully automated discovery and invocation. In the latter, techniques are evaluated by a subjective score issued by the organizers on the system design as well as on performance and accuracy.

IV. An empirical study of workflow discovery

In order to understand the practice and requirements of discovery we undertook two investigations: a survey and an analysis of Web logs of a workflow sharing platform. They provide insight on workflow sharing, re-use and discovery.

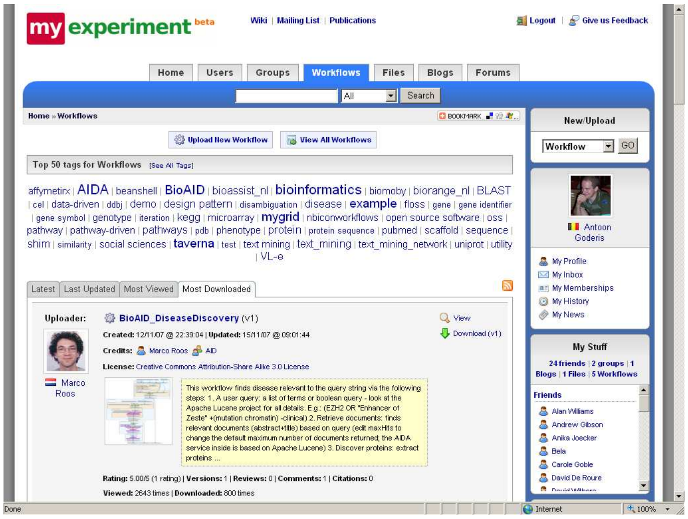
A. Empirical data sources used in the study

Both the survey and log analysis data are available on-line from www.myexperiment.org/benchmarks.

[4] Web sites: hub.inforsense.com and www.scitegic.com/products

[5] Web site: www.w3.org/TR/owl-features

[6] Web sites: ws-challenge.org and sws-challenge.org


my experiment beta
Wiki | Mailing List | Publications
Logout | Give us Feedback
Home
Users
Groups
Workflows
Files
Blogs
Forums
All
Search
Home » Workflows
BOOKMARK
Upload New Workflow
View All Workflows
Top 50 tags for Workflows [See All Tags]
affymetirx | AIDA | beanshell | BioAID | bioassist_nl | bioinformatics | biomoby | biorange_nl | BLAST | cel | data-driven | ddbj | demo | design pattern | disambiguation | disease | example | floss | gene | gene identifier | gene symbol | genotype | iteration | kegg | microarray | mygrid | nbiconworkflows | open source software | oss | pathway | pathway-driven | pathways | pdb | phenotype | protein | protein sequence | pubmed | scaffold | sequence | shim | similarity | social sciences | taverna | test | text mining | text_mining | text_mining_network | uniprot | utility | VL-e
Latest
Last Updated
Most Viewed
Most Downloaded
View
Download (v1)
New/Upload
Workflow
GO
Antoon Goderis
My Profile
My Inbox
My Memberships
My History
My News
My Stuff
24 friends | 2 groups | 1 Blogs | 1 Files | 5 Workflows
Friends
Alan Williams
Andrew Gibson
Anika Joecker
Bela
Carole Goble
David De Roure
Done
Internet
100%

Uploader: Marco Roos

BioAID_DiseaseDiscovery (v1)

Created: 12/11/07 @ 22:39:04 | Updated: 15/11/07 @ 09:01:44

Credits: Marco Roos AID

License: Creative Commons Attribution-Share Alike 3.0 License

This workflow finds disease relevant to the query string via the following steps: 1. A user query: a list of terms or boolean query - look at the Apache Lucene project for all details. E.g.: (EZH2 OR "Enhancer of Zeste" +(mutation chromatin) -clinical) 2. Retrieve documents: finds relevant documents (abstract+title) based on query (edit maxHits to change the default maximum number of documents returned; the AIDA service inside is based on Apache Lucene) 3. Discover proteins: extract proteins ...

Rating: 5.00/5 (1 rating) | Versions: 1 | Reviews: 0 | Comments: 1 | Citations: 0

Viewed: 2643 times | Downloaded: 800 times

Fig. 2. The myExperiment social networking site

1) Survey: From September to November 2007 we ran a survey published on Keysurvey.com. Twenty four bioinformaticians from 19 research laboratories participated. Of the 24 participants, 19 had built workflows before. Seventeen out of 19 were Taverna users. Survey questions and results are available on-line.

The survey was designed to document attitude towards sharing, re-use and discovery of workflows in the world of bioinformatics, where services can be local, private and under control of the author (e.g. a local database of microarray results) as well as distributed, public and autonomous (e.g. the NCBI BLAST analysis service at www.ncbi.nlm.nih.gov/blast).

2) Web log analysis of myExperiment.org: The myExperiment social network site, shown in Fig. 2, is designed to encourage workflow exchange [RGS07]. It is aimed to be workflow system neutral and currently provides support for Taverna and Triana (www.trianacode.org). myExperiment shares the Web 2.0 vision of sites such as mySpace or Flickr, but is set apart by being an exchange platform for scientific research objects. The latter brings in specific requirements in terms of managing ownership, credit, versioning, Intellectual Property, sharing incentives, permission policies, aggregation and provenance.

The site was launched in November 2007 and so far has received an average of 55 visits per day. Over the period January–April 2008, it counted 5647 visits coming from 78 countries. myExperiment currently has 751 users who contribute 150 unique workflows (excluding workflow versions).

We analysed the log files for the sharing, re-use and discovery patterns exhibited by users of the site. The log files span five months of activity (December 2007–April 2008). Due to privacy reasons, we cannot share the actual logs but instead report summary data for the analysis.

B. Insights into workflow sharing

1) Sharing attitude of the survey participants: Survey participants were asked about their *reservations about sharing* their workflows for re-use by others. Two main concerns came forward:

1) Receiving proper *acknowledgements* for the work (36.8%) and
2) The workflow doing the job, but not being a piece of software they are very proud of (31.6%).

Other factors were deemed less important:

- Being scooped (i.e. beat to obtaining results) by their own doing (15.8%);
- Sharing the data that is obtained from the workflows (15.8%);
- Sharing the data that feeds into the workflows (10.5%);
- The brittleness of shared workflows, either due to the use of non re-usable local services or due to the volatility of remote services (10.5%);
- Being able to share the workflow without others being able to establish how it works exactly (5.3%).

Our conclusion is that *participants are open to share quality workflows but want credit for doing so*. myExperiment caters for this attitude in part. It is designed explicitly to provide users with proper acknowledgements for their work. Mechanisms for workflow attribution, rankings of popular downloads and a community-based star rating system are available. The scientist has fine control over visibility and sharing of workflows, but other mechanisms to ease the fear of attracting a reputation as a poor workflow builder, such as "work in progress categories" or anonymous publishing, are not provided at this time.

2) Sharing attitude of the myExperiment community: Forty three out of 751 users (6%) share workflows on myExperiment. Of a total of 150 unique workflows (249 if all versions are counted), 40% of uploaders (2% percent of all users) contribute 80% of workflows.

The low percentage can be explained in different ways: (i) the same attitude as survey participants prevails but users have not yet shared their quality workflows or (ii) large user groups with different attitudes reside on myExperiment. Both options are plausible. For example, new users to Taverna registering on myExperiment during training days will only contribute with time – there is a lag effect. Similarly, the threshold for publishing may need to be lowered. In terms of (ii), we speculate there are "novelty-seeking" users with short attention spans, users dissatisfied with the site and users who, putting it in terms of behaviour exhibited on peer to peer networks, "leech." They would remain registered but not contribute.

C. Insights into workflow re-use

1) Re-use attitude of the survey participants: Polling participants about their concerns for workflow re-use, the following opinions surfaced:

- All respondents believed that in most cases there is not enough documentation to understand a workflow.
- For three quarters of respondents, some of the services in a workflow were (always or at least often) non-reusable due to the service being local to the original author. The same sentiment existed with respect to services being down.
- The majority of respondents believed there is no way of trusting the analysis performed by a workflow.

- Little under half of the respondents believed that often there are not enough workflows around, so they do not look for workflows.

Community-driven exchange platforms such as myExperiment can go a long way in meeting these concerns, through community-based annotation (description and tags), a repository of software code, workflow monitoring mechanisms and sharing best practice about building re-usable workflows.

One surprise finding, in light of the reported difficulty understanding workflows and trusting the analysis, is that of the 15 participants in the survey reporting re-use, seven had re-used workflows from third parties; other sources were fellow research group members (four mentions), project collaborators (four mentions) and a colleague at their institute (two mentions). The fact that *half of the re-users had adopted workflows from third parties* and not from people in their direct circle is an encouraging result for sites like myExperiment.

A second survey finding is that, again despite the difficulties understanding and trusting workflows, 15 out of the 19 workflow authors indicated having re-used workflows. The *high level of re-use activity* is remarkable. This may be due to the type of participant that volunteered to participate in the study – typically workflow enthusiasts and experts of Taverna who possess the skill set for successful re-use and repurposing.

2) Re-use attitude of the myExperiment community: Determining re-use attitude from the myExperiment logs is difficult. First, we need to decide on the appropriate metrics. To measure workflow re-use, we use:

1) The amount of *attributions* made on workflows regarding other workflows is a direct indication. So far, 12 attributions have been made on myExperiment.
2) The number of *workflow downloads* is an indirect indication. It is reasonable to assume a positive correlation between workflow downloads and the amount of workflow re-use by the downloader. During the measured period, 4216 workflows were downloaded. Download traffic from automated Web crawlers constituted 12%.

Second, the bulk of downloads are due to users who are not logged in, which makes it impossible to produce an accurate breakdown of downloads of users over time. For example, workflows may be found through Google or other interfaces to myExperiment content.

We found that *workflows published on myExperiment get downloaded*. Eighty two percent of workflows are downloaded five times or more. This fact supports the earlier speculation that myExperiment workflows are high quality. The most popular workflow, shown in Fig. 2, was downloaded 800 times (19% of total downloads).

D. Insights into workflow discovery

1) Discovery experiences of the survey participants: Ninety percent of respondents believed there are *no effective search tools to find relevant workflows.* The most quoted discovery mechanisms, in order of relevance, are: word of mouth, myExperiment and Google.

2) Discovery on myExperiment: myExperiment provides basic support for both content and context based discovery. In terms of content, an overall workflows Web page lists all entries. Detailed descriptions are harvested by analysing the uploaded workflows for textual descriptions and their constituting services. An internal search engine (SOLR) offers access to these through text queries. In addition, users add tags describing workflow contents, such as “phenotype” or “BLAST.”

In terms of context, multiple items are potentially associated with a myExperiment workflow, including user profile descriptions, public and private groups, files and attributing workflows. Some users provide tags containing contextual information, such as their organisation or project.

a) Effectiveness of the techniques: Given that we have no direct user feedback on the suitability of found workflows, we measure effectiveness through *the success of navigation and search in leading users to downloading workflows*. We measured which was the last retrieval action a user undertook before downloading a workflow, where retrieval actions consisted either of choosing a tag, issuing a search, clicking any user page, any group page or the list of workflows.

In total 4216 workflow were downloaded. *Navigation* of hyperlinks accounts for 56% of the total downloads, with tags at 30%, internal pages at 20% and incoming links from external pages at 6%. *Search* is responsible for 12% of downloads, with 6% from the internal search engine and 6% from Google. The remainder 32% come from direct downloads, of which a third are from Taverna users loading workflows directly into the workflow editor.

In terms of the tag cloud built by our community, we observe that 83% of tag-based downloads are triggered by 20% of tags. This confirms the well-known Pareto rule. Conversely, 63% of tags never lead to any workflow downloads.

b) Proportion of content-based versus context-based discovery: Given that both content-based and context-based search are in use, there exists a multi-modal approach to discovery on myExperiment. We wish to determine the relative importance of content and context during the retrieval process. This provides feedback on where to focus effort on discovery tools in future. The above techniques support both, e.g. the internal search engine accepts queries about people as well as biology.

To determine the relative share of content versus context, we analysed (i) the 280 tags in the tag cloud that lead to downloads, (ii) 927 textual queries fed to the internal search engine and (iii) the myExperiment Web pages that lead to downloads. Detailed figures are available on-line.

- For the *tag cloud*, we classified tags into *content* categories (Biology, Bioinformatics, Workflow technology, Other sciences), *context* categories (People, Organisation, Project, Event, Place and *varia* (includes Other and Unknown topics). It turns out at least 55 % of tags used for downloading were content based and at least 26 % is context based (19% were varia).
- For the *queries*, we could keep the same subcategories as above for content, context and varia. At least 69 % of queries were content based and at least 9 % were context based (22 % were varia).
- Within the myExperiment *Web pages*, 92 % of downloads

stemmed from the workflow list page, 5% from a user page and 3% from a group page.

We observe that *content-based discovery dominates across the board but equally there is a role for context-based discovery*. In future analysis, investigating the navigation path in closer detail may reveal more about the interplay of context-based and content-based discovery.

V. Benchmarking workflow discovery

The perception of survey participants that no effective discovery tools exist motivates experimentation with novel workflow discovery methods. We established earlier that almost no work exists on evaluating workflow discovery techniques. Here we present our work towards building a set of benchmarks.

To construct a benchmark for workflow discovery, one option was to use empirical data directly from myExperiment. The major problem with log based approaches is that the user information need is not explicitly captured. For example, during the analysis of navigational paths, it is difficult to determine at which point in the traversed path the scientist starts to think about her particular discovery problem at hand and to determine what that problem is.

Instead, we constructed benchmarks derived from *controlled experiments*. They are specific to the case of *content-based* discovery of *finished concrete* workflows, where workflows are searched for *by example*.

A. Overview

We conducted three small-scale controlled experiments and one larger one. They rely on a corpus of real-world bioinformatics workflows, as generated by domain experts with the Taverna workbench. The experiments differ widely in their setup, reflecting the different approaches taken for capturing how re-use occurs, the different conditions under which re-use occurs and practical constraints involved in running user experiments. Table I provides an overview of the experiments according to their experimental setup, participants, materials, procedure and results. We discuss the details below. A more detailed technical report, the list of participants, the materials used and benchmark data are available from www.myexperiment.org/benchmarks.

B. Experimental setup

1) Re-use and discovery tasks measured: In designing experiments to capture a user's re-use and discovery behaviour one has to be selective in what is measured.

	1	2	3	4
Setup				
Re-use subtasks	discovery, editing	discovery	discovery	discovery, editing
Discovery task	sub, super	earlier versions	sub, super, alternative	sub, super
Re-use directions	B2C	A2A	B2A	B2A, A2B, B2C
Participants				
Number	15	2	2	24
Expertise	++	++++	++++	+++
Materials				
Workflows	20	67+78	33	18
Documentation	++	++	++	++++
Procedure	embedded	independent	independent	independent
Results				
Average duration	20 min.	105 min.	30 min.	90 min.
Statistical support	-	N/A	+++	+++
Assessments	N/A	145	456	1848

TABLE I
Overview of the user experiments.

In earlier work [GLG06], we attempted in vain to capture universal metrics that bioinformaticians use to establish similarity between workflows (e.g. the number of services shared *and* not shared between workflows). Such metrics could then be confidently incorporated in discovery tools to support a workflow by example discovery approach. In hindsight, we believe that the negative outcome was mostly due to workflow discovery being driven by a concrete information need. People are known to approach similarity based on multiple cognitive approaches [Gol01]. Leaving the purpose for a similarity measurement unspecified leaves participants with many options to base similarity assessments on.

The controlled experiments presented here fix the user information need by giving participants clear re-use and discovery goals. Experiments 2 and 3 aimed to record discovery behaviour, while experiments 1 and 4 also captured edit behaviour.

Experiment 1 asked for discovery of workflows that do either a supertask or a subtask of a provided exemplar workflow, and to select relevant fragments in them.

Experiment 2 set the task as identifying all versions of a workflow authored by the participant within a wider set of his workflows.

Experiment 3 asked for boolean assessments of "usefulness." Useful was defined as meaning that one workflow either (i) provides an alternative to the other, (ii) provides an extension to it (supertask) or (iii) provides a useful fragment of it (subtask).

Experiment 4 gave participants practical re-use and repurposing tasks. Consider Fig. **??** in the appendix, which shows one of the twelve exercises shown to participants (shrunk from its original A1 sized format). The challenge is to to solve the task stated at the bottom by adapting the workflow with the circled number 1, using drawings on the poster. The particular task in question is to discover and then edit the workflow(s) which enable the given workflow to obtain a list of gene

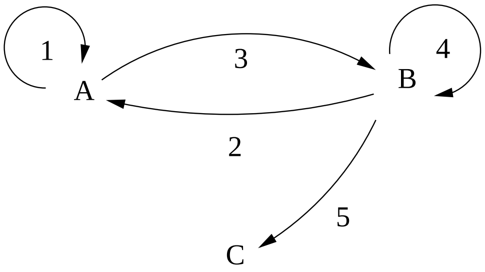


Fig. 3. Types of workflow re-use from the perspective of the author of a set of workflows A.

identifiers by simplifying its BLAST output file. The solution is provided here in the figure. Note that we did not inform participants which edit operation to undertake (in this case, an extension). After solving the tasks, participants were asked about the way they had combined the discovery and editing steps.

2) Re-use directions: We took into account the fact that workflows are re-usable between several parties in several directions. It is important to make these distinctions because they are expected to influence the difficulty of a re-use task and the strategies people use to solve it, e.g. re-use of a set of workflows one is familiar with should be easier.

We distinguish between *personal re-use*, involving only workflows authored by the re-user, and *cross-author re-use*, involving other workflows. Figure 3 summarizes the five possible ways a user can re-use her own as well as other people's workflows. Assume she has a number of workflow sources available: her own set of workflows **A**, a set of workflows **B** created by a second party and a set of workflows **C** created by a third party.

1) **Personal re-use: from A to A (A2A)** Re-use of workflows from a personal collection, which are either versions of the workflow one is currently working on, or previously built workflows with a different topic altogether.
2) **Cross author re-use: from B to A (B2A)** Re-use of workflows from someone else's collection to alter one's own current workflow.
3) **Cross author re-use: from A to B (A2B)** Re-use of workflows from one's own collection to alter a workflow from someone else's collection.
4) **Cross author re-use: from B to B (B2B)** Re-use of workflows from someone else's collection to alter a workflow from that same collection.
5) **Cross author re-use: from B to C (B2C)** Re-use of workflows from someone else's collection to alter a workflow from another external collection.

Table I summarises which experiments investigate which re-use directions. Interestingly, for experiment 4, which direction is measured in a given re-use exercise is dependent on the combination of a particular re-use task and the participant in question. For example, the case where none of the workflows in the exercise are authored by a participant means either B2B and B2C is being measured. If the original author of one of the workflows is solving that same task, one would be measuring A2A, A2B or B2A instead.

C. Participants

Between two and 24 bioinformaticians participated in any given experiment. In experiment 1 participants had no workflow experience. In experiments 2 and 3, we used the same two authors, both of whom had authored over 100 bioinformatics workflows. Experiment 4 drew on the survey participants, where 19 out of 24 had workflow experience.

D. Materials

Workflows were chosen according to the function of the re-use task to be measured (e.g. finding versions). Conversely, the characteristics of our workflow corpus influenced the setup of re-use tasks. For instance, for experiment 4 we invested two man months in the annotation of workflows.

The different experiments showed different amounts of workflow detail to users. In experiments 1 to 3, a workflow's name was shown as well as the orchestration of its services rendered as a diagram, showing only those inputs and outputs actively involved in the orchestration (as in Fig. 1). In experiment 4, more detail was provided with the inclusion of textual descriptions of the overall workflow task and of the services. In addition, semantic annotation was provided describing the task, inputs and outputs of the 98 services present in the 18 workflows, based on concepts selected from the ^{my}Grid bioinformatics service ontology.[7]

Workflows were presented to participants on paper – A4-format for experiments 1-3, A1-format for experiment 4.

E. Procedure

Organisation-wise, the first experiment piggybacked on a training day for the Taverna workflow editor. The others were set up independently.

For experiment 4, the exercises were first tested in two pilot studies with two post-doctoral bioinformaticians, leading to changes in the vocabulary used in the instructions, the curation and the re-use task descriptions. A third bioinformatician verified the validity of what consisted the correct solutions to the tasks, by creating and testing the corresponding workflows in Taverna for all tasks. For experiments 3 and 4 participant agreement was calculated based on Kappa measures [SC88].

F. Results

Following our experimental setup, both quantitative and qualitative results were generated.

1) Quantitative results:

a) A set of content-based benchmarks: The work of participants translated into a documented set of decisions made during the workflow re-use process. The outcome of each experiment was judged to be positive only when the results from the exercises showed a level of agreement between participants and were confirmed by a bioinformatician as being sensible. In experiment 1, the combination of inexperience with workflows and poor quality workflow descriptions resulted in demotivated participants, who gave up rapidly on

[7] Navigateable at www.mygrid.org.uk/ontology/OwlDoc/index.html

Benchmark	Exp.	Participants	Behaviour captured	Assessments	Participant agreement (Kappa value)
PR	2	2	Personal discovery	145	N/A
CA2	3	2	Cross-author discovery	456	Very good (0.678)
CA24	4	24	Cross-author repurposing	1848	Very good (0.666)

TABLE II
Overview of content-based benchmarks

the task and yielded no useful answers. The three other user experiments had positive outcomes and produced benchmarks with different characteristics. They are named after the type of re-use captured and the number of participants.

Benchmark **PR2** (experiment 2) collects similarity assessments made by a workflow author about pairs of his own workflows. In Benchmark **CA2** (experiment 3), a collaborator made similarity assessments on those same workflows. Benchmark **CA24** (experiment 4) contains the assessments made regarding the relevance of candidate workflows to solve specific tasks.

b) Participant confidence and agreement: All benchmarks are created by participants who felt confident while creating them. For **PR2** and **CA2**, both participants felt confident and agreed strongly on the assessments made, as shown by the Kappa statistic for inter-rater agreement (see Table II). Agreement was never perfect, though.

The same is true for **CA24** (experiment 4). Analysis of ratings shows participants in general had high confidence and found the exercises to be of easy to moderate difficulty. Surprisingly, analysis of inter-rater agreement showed that they did not agree which exercises were easy, moderate or difficult. Similarly, they did not agree when they had high, medium or low confidence. An explanation for this apparent paradox is either that participants come from very different backgrounds and thus find different tasks challenging, or they use a different internal scale to assess confidence and difficulty. Their results on relevance assessments suggest the latter is true.

On the other hand, participants of **CA24** did agree on the relevance assessments made – a multi-rater Kappa value of 0.666 was obtained. Again agreement was not perfect. For this benchmark, because the correct answers were field-tested, disagreement on relevance assessments could be measured in terms of correctness. Contrasting participant relevance assessments with the correct solution shows that they on average were correct in 83% to 91% of all cases, depending on the scheme adopted to assess a given answer. The schemes vary on:

- whether they should count a "maybe" answer as a correct answer (which leads to better scores) or whether it should be excluded from the performance measure.
- whether blank answers should count as negative answers (which leads to better scores, given that the majority of candidate workflows are not relevant to a particular task) or instead should be excluded.

The main sources of error for participants in **CA24** were (i) incomplete exercises because of a natural "*blind spot*" in the exercise material, (ii) incomplete or ambiguous *descriptions of data items*, (iii) assumptions made on the required *generality of a solution* across species, and (iv) assumptions made on the *admissibility of additional "shim" or glue services* which were not available from any of the presented workflows. We also analysed whether the amount of *expertise* building workflows or the *time* taken to complete the exercise showed a correlation with the level of *correctness* obtained. Neither factor proved to be a determinant. This indicates that people with a good bioinformatics background in general can muster the tasks of editing workflow diagrams and that some people simply work faster than others.

2) Qualitative results:

a) Bioinformaticians are capable of all types of workflow discovery when the conditions are right: The commonsense expectation is that participant familiarity with the workflow author, participant motivation and participant expertise correlate positively with valid answers to discovery tasks. This expectation was confirmed in all experiments.

b) The relative impact of documentation on workflow re-use and discovery was uncovered: Lots of quality workflow documentation is no requirement to achieve good results when it comes to discovery of one's own workflows or workflows by collaborators, as shown by experiments 2 and 3. Experiment 4 showed that the combination of motivation, expertise and quality metadata enables discovery from external parties. In contrast, the combination of inexperience with workflows and poor quality workflow descriptions in experiment 1 resulted in severe demotivation of participants, who gave up rapidly on the task and produced no valid answers. We hypothesise that *documentation plays a crucial role* for successful re-use either indirectly (to drive motivation) or directly (to inform the discovery process).

c) Understanding of workflow re-use and discovery behaviour: Experiment 4 had the ambition to model bioinformatician behaviour, in particular the assessment of relevance of potential workflow candidate solutions and their subsequent editing. Results show that *relevance assessment and editing are done in two distinct phases*. First, participants scan the whole population of available workflows. After this, editing is done on the workflows marked as relevant. It also documents *which sources of information are used in which phase*. For both phases, the workflow diagram was the first and most used point of recourse for finding information, despite its low detail and ambiguity. This finding underlines the power of using a visual medium. Textual workflow and service inputs and outputs were also used eagerly in both phases, but less so than the diagram. The overall workflow description and workflow name were deemed useful for relevance assessment only.

VI. Evaluating against the content-based benchmark

A. Setup

The developed suite of benchmarks captures re-use behaviour in data flows. Each scientific workflow system capable of modelling data flows should be able to re-model the workflows used into its own language, provided equivalent services are available. It could then test its own discovery system with respect to the benchmarks. The fact that the workflows are from bioinformatics should not matter, provided the discovery system in place is domain independent.

To test the benchmark data against real tools, we selected two techniques specifically developed for Taverna Scufl workflows. Details of Taverna's workflow language are in [OGA$^+$05]; see Fig. 1 for two examples. The first tool is an existing graph matching based tool [GLG06]. The second tool is new and consists of an adaptation of the Woogle search engine for Web services [DHM$^+$04].

Graph matching over Taverna workflows Graph matchers assume graphs of a certain kind as input; in the case of [GLG06], the graph matcher works over attribute-less graphs of nodes and directed, attribute-less edges. To produce results, the graph matcher relies on sub-isomorphism detection over a graph repository.

The content of a graph impacts the outcome of the matching process. The translation from workflow to graph was done as follows. The workflow's overall inputs and outputs are included as named nodes in the graph. The intermediate nodes are instantiated with the names of the services connecting the workflow's input and output, while ignoring all information about intermediary inputs and outputs. The graph's edges are defined as the connections between the services.

Text clustering over Taverna workflows Workflows are not only software specifications. They are also documents which contain natural language. One can therefore apply information retrieval on workflow descriptions. Woogle is a tool for similarity search for Web services that relies on standard information retrieval techniques as well as its own clustering algorithm.

To adapt the tool to workflows, we abstract a workflow to be a bag of services. Essentially we establish a lossy translation of a workflow into the format of a Web service. We wrote a parser to translate Scufl workflows into the Woogle WSDL service input format by regarding each workflow as a WSDL service and each constituent workflow service as a WSDL operation. The technique takes in a collection of Scufl workflows, clusters these in an off-line step, and then, when given an input workflow, produces rankings of workflows from the collection.

In addition to the raw performance of these two techniques, we also consider the "combination hypothesis" as an additional technique – the idea that further advances in search technology will be based on a cross-disciplinary approach. In our context, we consider the impact of combining the results of the graph matching and text clustering techniques. We identify two options: (i) use the *intersection* of results (when both techniques agree) or (ii) use the `union` of results.

Measure	Top *x* results	Graph matcher	Text clustering	Intersection	Union
Precision	25	65	34	51	44
Recall	25	50	24	17	57
Precision	10	65	35	90	48
Recall	10	21	9	7	25
Precision	5	70	40	83	56
Recall	5	12	6	2	16

TABLE III
Average recall and precision on **PR2**.

Measure	Top *x* results	Graph matcher	Text clustering	Intersection	Union
Precision	11	-	60	-	-
Recall	11	-	74	-	-
Precision	5	-	50	-	-
Recall	5	-	36	-	-

TABLE IV
Average recall and precision on **CA2**.

B. Evaluation results

We test the graph matcher and the text clustering tool on benchmarks 1 and 2.

Table III summarises the performance of the 2+2 techniques for personal re-use. It shows the average precision and recall for performing the versioning tasks. Table IV gives the average precision and recall for cross-author discovery for the 11 workflows used as basis of comparison. It shows performance with respect to the top *x* results returned by a given technique (values in percentage; higher is better).

The figures bring out the trade-off between precision and recall, in that an increase in precision means a decrease in recall. The only exception to this is the performance of the text clustering, which might be explained by the relative small set of 21 workflows over which the clustering algorithm operated.

The different classes of discovery techniques come with their own strengths and weaknesses. The *text clustering* technique performs well on cross-author discovery, but does poorly when it comes to versioning. The *graph matcher* does well in comparison on the versioning task. When applying the graph matcher for cross-author discovery, however, no results are returned in any of the cases. Inspection of results revealed its lack of a lexical component is to blame. As a result, the application of the combination hypothesis turns out to be sensible only in the case of versioning, where both techniques yield results. The *intersection* technique has good precision on the versioning task compared to the other techniques, but displays a drop in recall, whereas the *union* technique displays a converse pattern. We conclude that the combination hypothesis idea does not improve the quality of search results overall in our experiment; one has to choose between either bettering precision or bettering recall.

By using the benchmark we have shown that these techniques alone do not approach what humans can achieve or would expect of the system. Comparing the results of the techniques and the experts, we found multiple matches which were only identified by the experts. These missed matches relied on expert background knowledge of the biology and bioinformatics behind the services.

VII. Conclusion and future work

Workflows are proving successful in automating scientific experiments conducted on the Web. Public repositories are appearing to enable their re-use and repurposing into new experiments.

We investigated current practices in workflow sharing, re-use and discovery amongst life scientists chiefly using the Taverna workflow management system and the myExperiment workflow exchange platform. In terms of sharing and re-use, we find that (i) an enthusiastic core is willing to share quality workflows but expects credit for doing so. They act as provider to the wider community; (ii) workflows that are shared get re-used; (iii) workflow re-use is perceived as hard but doable.

In terms of discovery, we find that (iv) both workflow content and workflow context is important in supporting discovery; (ii) the perception is that no effective discovery tools exist; (iii) a range of specialised techniques exists, unexploited by the scientific workflow community; (iv) no means are available to systematically evaluate these.

This paper has demonstrated a methodology for evaluating workflow discovery tools that is not specific to the particular science domain nor the choice of workflow system. We successfully built a benchmark measuring the re-use and discovery behaviour of life scientists. We showcased the evaluation of two Taverna-based techniques.

Further empirical work can be undertaken to elicit discovery patterns. Ongoing parallel studies appear to corroborate our own findings. More studies can be conducted based on the benchmarks with tools, potentially from other workflow environments. Additional benchmarks can be devised, for example to record how scientists manipulate scientific research objects that encapsulate workflows and link to other data. In this respect, myExperiment itself is a powerful instrument for measuring sharing, re-use and discovery behaviour.

Acknowledgment

The authors are grateful to the February 2006 Taverna training day participants and the ^{my}Grid team. We also thank Andy Gibson, Peter Li, Katy Wolstencroft and the participants of the survey.

References

[BK02] Abraham Bernstein and Mark Klein. Towards high-precision service retrieval. In *Proceedings of the First International Semantic Web Conference (ISWC)*, Sardinia, Italy, 2002. Springer.

[BKBK05] A. Bernstein, E. Kaufmann, C. Brki, and M. Klein. How similar is it? towards personalized similarity measures in ontologies. In *7 Internationale Tagung Wirtschaftsinformatik*, February 2005.

[BYRN99] Ricardo Baeza-Yates and Berthier Ribiero-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[CGB06] J. C. Corrales, D. Grigori, and M. Bouzeghoub. Bpel processes matchmaking for service discovery. In *Conference on Cooperative Information Systems (COOPIS)*, LNCS 4275, pages 237–254, Montpellier, France, 2006.

[DG06] Ewa Deelman and Yolanda Gil. Managing large-scale scientific workflows in distributed environments: Experiences and challenges. In *Second IEEE International Conference on e-Science and Grid Computing*, Amsterdam, December 4-6 2006.

[DHM$^+$04] X. Dong, A. Halevy, J. Madhavan, E. Nemes, and J. Zhang. Similarity search for web services. In *Proc.s of the 30th VLDB Conference*, Toronto, Canada, 2004.

[GBA$^+$07] A. Goderis, C. Brooks, I. Altintas, E. A. Lee, and C. Goble. Composing different models of computation in kepler and ptolemy ii. In *Proc. of Int. Conference on Computational Science (ICCS) 2007*, Beijing, China, May 27-30 2007.

[GDE$^+$07] Yolanda Gil, Ewa Deelman, Mark Ellisman, Thomas Fahringer, Geoffrey Fox, Dennis Gannon, Carole Goble, Miron Livny, Luc Moreau, and Jim Myers. Examining the challenges of scientific workflows. *Computer*, 40(12):24–32, December 2007.

[GHS95] Dimitrios Georgakopoulos, Mark F. Hornick, and Amit P. Sheth. An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3(2):119–153, 1995.

[GL05] C. Goble and B. Ludaescher, editors. *ACM Sigmod Record: Special Issue on Scientific Workflows*, volume 34. September 2005.

[GLG06] Antoon Goderis, Peter Li, and Carole Goble. Workflow discovery: the problem, a case study from e-science and a graph-based solution. In *IEEE Int. Conf. on Web Services*, Chicago, USA, September 18-22 2006.

[Gol01] R. L. Goldstone. *MIT encyclopedia of the cognitive sciences*, chapter Similarity, pages 757–759. MIT Press, Cambridge, MA, 2001.

[KBL$^+$07] Christoph Kiefer, Abraham Bernstein, Hong Joo Lee, Mark Klein, and Markus Stocker. Semantic process retrieval with iSPARQL. In *European Semantic Web Conference (ESWC)*, pages 609–623, 2007.

[MW06] Bendick Mahleko and Andreas Wombacher. Indexing business processes based on annotated finite state automata. In *ICWS*, pages 303–311, 2006.

[OGA$^+$05] Tom Oinn, Mark Greenwood, Matthew Addis, Nedim Alpdemir, Justin Ferris, Kevin Glover, Carole Goble, Antoon Goderis, Duncan Hull, Darren Marvin, Peter Li, Phillip Lord, Matthew Pocock, Martin Senger, Robert Stevens, Anil Wipat, and Chris Wroe. Taverna: Lessons in creating a workflow environment for the life sciences. *Concurrency and Computation: Practice and Experience: Special Issue on Scientific Workflows*, 2005.

[RGS07] David De Roure, Carole Goble, and Robert Stevens. Designing the myexperiment virtual research environment for the social sharing of workflows. In *Third IEEE International Conference on e-Science and Grid Computing*, pages 603–610, Bangalore, India, December 10-13 2007.

[SC88] S. Siegel and J. N. Castellan. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, 1988.

[SVK$^+$07] Carlos E. Scheidegger, Huy T. Vo, David Koop, Juliana Freire, and Claudio T. Silva. Querying and creating visualizations by analogy. *IEEE Trans. Vis. Comp. Graph.*, 13(6):1560–1567, 2007.

[WGG$^+$07] Chris Wroe, Carole Goble, Antoon Goderis, Phillip Lord, Simon Miles, Juri Papay, Pinar Alper, and Luc Moreau. Recycling workflows and services through discovery and reuse: Research articles. *Concurr. Comput. : Pract. Exper.*, 19(2):181–194, 2007.

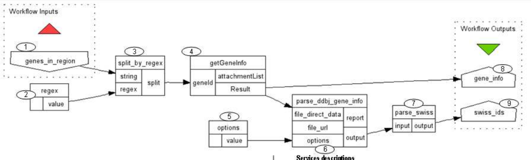
[Wom06] A. Wombacher. Evaluation of technical measures for workflow similarity based on a pilot study. In *CoopIS*, Montpellier, France, November 1-3 2006.

[WSG$^+$03] C. Wroe, R. Stevens, C. Goble, A. Roberts, and M. Greenwood. A suite of daml+oil ontologies to describe bioinformatics web services and data. *Intl. J. of Cooperative Information Systems*, 12(2):197–224, 2003.

[ZWF06] Y. Zhao, M. Wilde, and I. Foster. Applying the virtual data provenance model. In *Int. Provenance and Annotation Workshop (IPAW)*, Chicago, USA, May 3-5 2006.

Appendix
Example exercise used in experiment 4

Example exercise



Overall workflow description "Ensembl_id_2_Swissprot_id.xml"

This workflow extracts gene information and the relevant swissprot ids given Enembl gene ids

Inputs
① genes_in_region: List of Ensembl gene ids.

② regex: Regex value to use for "split_by_regex" operation

⑤ options: option value used to extract a piece of data from "parse_ddbj_gene_info" output file. e.g. swiss

Outputs
⑧ gene_info: return gene information

⑨ swiss_ids: return swissprot ids

Services descriptions

③ **Split_by_regex (Split string into string list by regular expression)**: split a given string with a specified regular expression (regex)
Input
String: string to split
Regex: regular expression
Output
split: return split string

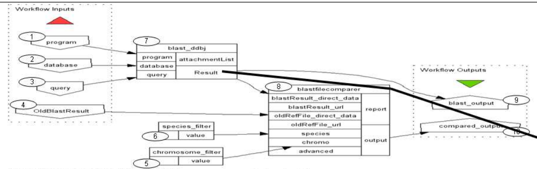
④ **getGeneInfo**: retrieves gene information given a Ensembl gene id [retrieving]
Input
geneId: Ensembl gene id [ensembl_record_id]
Output
Result: Return gene info of specified Ensembl gene id [Ensembl_record]

⑥ **Parse_ddbj_gene_info**: extract information from DDBJ (Dna Data Bank of Japan) getGeneInfo processor [retrieving]
Input
file_direct_data: 'getGeneInfo' output result [Ensembl_record]
option: used to extract a piece of data from output file. e.g. swiss
Output
Output: return the extracted piece of data

⑦ **parse_swiss**: Beanshell script to extract only swissprot id from "parse_ddbj_gene_info" output Record.
Input
input: parse_ddbj_gene_info output record with 'swiss' as option.
Output
output: Return swissprot ids [SWISS-PROT_accession]

6

Relevant? Yes [ ] No [X] Maybe : ______________________________



Overall Workflow Description "BlastComparer.xml"

This workflow performs a BLAST search then compares the result to a previous blast result based on specified filter.

Inputs:
① program: blasttype: blastn, blastp, blastx, tblastn or tblastx

② database: e.g. SWISS, NCBI, EMBL, DDBJ

③ query : nucleotide or protein sequence

④ OldBlastResult: blast result

⑥ species_filter: species name

⑤ chromosome_filter: chromosome number

Outputs:
⑨ blast_output: result of blast execution

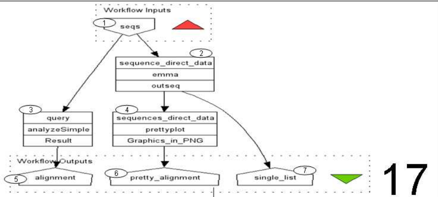
⑩ Compared_output: return a list of GI number

Services descriptions

⑦ **blast_ddbj (searchSimple):** Executes BLAST with specified program, database and query [local_aligning]
inputs:
program: Specify blast type used: blastn, blastp, blastx, tblastn or tblastx
database: Specify database: eg. SWISS, NCBI, EMBL, DDBJ . For all possible databases: *see appendix*
query: nucleotide or protein sequence [biological_sequence]
Output
Result: result of blast execution [BLAST_report]

blastfilecomparer: Compares a new BLAST output to an older blast output to identify new hits [filtering]
⑧ Inputs
blastResult_direct_data: blast result file.[BLAST_report].
Use either this parameter or blastResult_url parameter as input but not both together.
blastResult_url: url of the blast result. .[BLAST_report].
oldRefFile_direct_data: old blast result file..[BLAST_report].
Use either this parameter or oldblastResult_url as input but not both together.
oldRefFile_url: url of the old blast result
species: filter the result by species name
chromo: filter the result by chromosome number
advanced: words are looked for in the FASTA definition line
Output
report: Return a filtered blast result.

1



Overall workflow description "simpleAlign.xml"
This workflow aligns given sequences and displays aligned sequences, with colouring and boxing.

Input
① seqs: nucleotide or protein sequence in fasta format

Outputs
⑤ alignment: return sequence alignment result using analyzeSimple operation

⑦ single_list: return sequence alignment result using "emma" operation

⑥ pretty_alignment: Return alignment result with colouring and boxing.

Services descriptions

② **emma:** Multiple alignment program - interface to ClustalW program [aligning]
Input
sequence_direct_data: nucleotide or protein sequence [biological_sequence]
Output
outseq: Return aligned sequence [multiple_sequence_alignment_report]

③ **analyseSimple:** Execute ClustalW specified with multi sequences [aligning].
Input
query: nucleotide or protein sequence [biological_sequence]
Output
result: Return aligned sequences [multiple_sequence_alignment_report]

④ **prettyplot:** Displays aligned sequences, with colouring and boxing [displaying]
Input
sequence_direct_data: File containing a sequence alignment [multiple_sequence_alignment_report] [pairwise_sequence_alignment_report]
Output
Graphics_in_PNG: Return a plot of aligned sequences.

Relevant? Yes [ ] No [X] Maybe : ______________________________

Based on workflow (1), obtain a list of gene identifiers by simplifying the BLAST output file.

Difficulty of task

Difficult [ ] Moderate [X] Easy [ ]

If difficult, please explain: ______________________________

Confidence level in solution

High [X] Medium [ ] Low [ ]

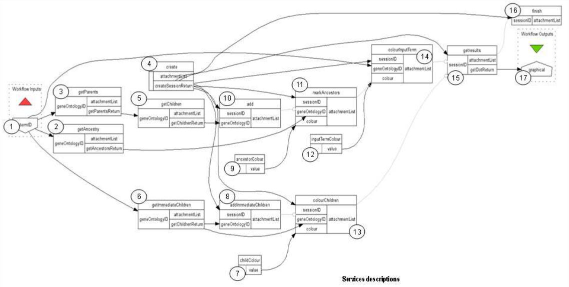
If low, please explain: ______________________________

Please indicate the workflows (if any) where the diagram **alone** provides enough information to determine whether it is a solution **or not**.

12: nothing involving BLAST 2: question is in workflow

Please indicate the workflows (if any) where the [semantic tagging] provides **essential** information to determine whether it is a solution **or not**.

wf17:service3 "Result" is mult. seq. alignment report, not BLAST report



Overall workflow description "ShowGeneOntologyContext.xml"

This workflow builds up a sub graph of the Gene Ontology given a GO term id to show the context for a supplied term or terms

Inputs

① termID: GO term id. e.g. GO:0007601

⑦ childColour: colour to use for specify children

⑨ ancestorColour: colour to use for specify ancestors

⑫ colourInputTerm: specify the colour of given terms.

Outputs

⑰ graphical: Return a sub graph of the Gene Ontology given a GO id.

Services descriptions

② **getParents:** Retrieves the IDs of all immediate parent terms of specified GO ID [retrieving]
Input
geneOntologyID: GO ID of which the Parent terms should be returned [Gene_Ontology_term_id]
Output
getParentsReturn: Return the IDs of all immediate parent terms of the specified term [Gene_Ontology_term_id]

③ **getAncestry (getAncestors):** Retrieves the IDs of all ancestors of specified GO ID [retrieving].
Input
geneOntologyID: GO ID of which the Parent terms should be returned [Gene_Ontology_term_id]
Output
getAncestorsReturn: Return the IDs of all ancestors of the specified term [Gene_Ontology_term_id].

④ **Create (createSession):** Takes no arguments and Creates a new GoViz session on the server and returns a session identifier that can be used in subsequent operation.
Output
createSessionReturn: Return a session identifier that can be used in subsequent operation.

⑤&⑥ **getChildren & getImmediateChildren (getChildren):** Retrieves the IDs of all immediate children of a specified GO ID [Retrieving].
Input
geneOntologyID: GO ID of which the Children should be returned [Gene_Ontology_term_id].
Output
getChildrenReturn: Return the IDs of all immediate children of the specified term [Gene_Ontology_term_id].

⑧&⑩ **addImmediateChildren & add (addTerm):** Add a GO term to the visualisation, updating the state of the named session.
Input
SessionID: Session ID returned by the createSession operation.
geneOntologyID: GO ID of which the Parent terms should be returned [Gene_Ontology_term_id]

Services descriptions

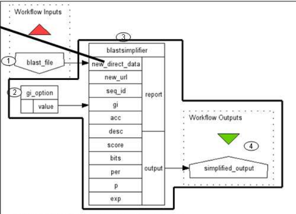
⑪&⑬&⑭ **markAncestors & colourChildren & colourInputTerm (markTerm):** Adds a specific colour parameter to supplied term in the Gene ontology.
Inputs
SessionID: Session ID returned by the createSession operation.
geneOntologyID: GO ID of which the Parent terms should be returned [Gene_Ontology_term_id]
colour: The colours can be anything that is a valid colour within the dot file format. For the list of colours see appendix:

⑮ **getresults (getDot):** Retrieves the DOT text specifying the sub graph of the Gene Ontology that contains all the terms that have been added to the session. [retrieving]
Input
sessionID: SessionID: Session ID returned by the createSession operation.
Output
Return the DOT text specifying the subgraph of the Gene Ontology.

⑯ **Finish (destroySession):** Removes a session from the server, identified by the session ID returned by the createSession operation.
Input
SessionID: Session ID returned by the createSession operation.

8

Relevant? Yes [ ] No [X] Maybe : ______________________________



Services descriptions

③ blastsimplifier: Simplifies BLAST output by specifying elements (seq_id, gi, acc, desc, Score, bits, per, p, exp) to be displayed in the blast result output. [filtering].

Inputs
new_direct_data: blast report file [BLAST_report].
mutually exclusive with new_url parameter
new_url: url of the blast report file [BLAST_report].

The following parameter are optional. To select one of them , pass the name of the input as input parameter. For example to display GI numbers, pass gi to the parameter gi.

seq_id: sequence identifier

gi: For GI number

acc: For accession number

desc: for descriptions

score: for score value

bits: for bits score

per: for percentage of identity.

p: for p-value

exp: for E-value

Output
report: return a simplified blast report

Overall Workflow Description "blast_simplifier.xml"

This workflow simplifies a BLAST textfile into identifiers, descriptions and values (P, E-values).
In order to extract the relevant ids etc. you need to pass the relevant string into the corresponding port, e.g. the default port being used is gi. This has been passed "gi". For any other ports simply pass in the string the SAME as the port name, e.g. seq_id, p, per etc.

Inputs

① blast_file: blast result

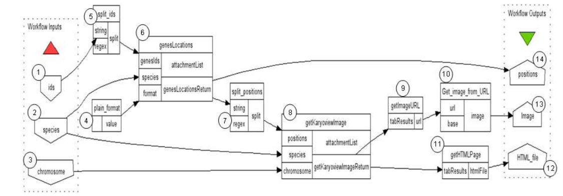
② gi_option: here we want to retrieve only the gi number from the blast output.

Outputs

④ Simplified_output: list of GI numbers

2

Relevant? Yes [X] No [ ] Maybe : ______________________________



Overall workflow description "Karyoview.xml"
This workflow retrieves and displays genes positions on a chromosome using Ensembl Karyoview.

Inputs

① ids: list of gene id. e.g. BRCA2, ENSG00000128573

② species: species name. e.g. homo sapiens

③ chromosome: chromosome number.

④ plain_format: format of the gene id list

Outputs

⑫ HTML_file: HTML file of the URL containing the image

⑬ image: image of the genes positions on the chromosome

⑭ Position: position of gene on the chromosome

Services descriptions

⑨ **getImageURL**: Beanshell script, extracts the URL of the karyotype
Input
tabResult: result of the "getKaryoviewImage" service
Output
url: return the URL of the karyotype.

⑪ **getHTMLPage:** Beanshell script, extracts the html page of the karyotype
Input
tabResult: result of "getKaryoviewImage" service
Output
HTMLPage: Return the HTML page of the karyotype.

⑩ **Get_image_from_URL (Get image from URL):** retrieves the image given the URL
input
url: URL of the image
Output
Image : Return the image of specified URL

Services descriptions

⑤ **Split_ids (Split string into string list by regular expression):** split a given string with a specified regular expression (regex)
Input:
String: string to split
Regex: regular expression (here:"\n")
Output:
split: return split string

⑥ **genesLocations:** retrieves the location of a gene on a genome using its identifier [retrieving]
Inputs
genesIds: gene identifier .e.g. ENSG00000128573 [Ensembl_record_id]
species: species name. e.g. homo sapiens
format: format of the gene id list. e.g. plain
Output
genesLocationsReturn: Return the location of genes on the chromosome.

⑦ **split_positions (Split string into string list by regular expression):** split a given string with a specified regular expression (regex)
Input:
String: string to split
Regex: regular expression (here:"\n")
Output:
split: return split string

⑧ **getKaryoviewImage:** Returns a representation of the karyotype of a species with features we want to locate on [displaying]
Inputs
position: position of genes on the chromosome
species: species name. e.g. homo sapiens
chromosome: chromosome number
Outputs
getKaryoviewImageReturn: return the URL and html file of karyotype

12

Relevant? Yes [ ] No [X] Maybe : ______________________________