SHARP: Harmonizing and Bridging Cross-Workflow Provenance

Alban Gaignard, Khalid Belhajjame, Hala Skaf-Molli

SHARP: Harmonizing and bridging cross-workflow provenance

Alban Gaignard[1], Khalid Belhajjame[2], and Hala Skaf-Molli[3]

[1] l'institut du thorax, INSERM, CNRS, UNIV Nantes, Nantes, France
alban.gaignard@univ-nantes.fr
[2] Université de Paris-Dauphine, LAMSADE, Paris, France
kbelhajj@googlemail.com
[3] Université de Nantes, LS2N, Nantes, France
hala.skaf@univ-nantes.fr

Abstract. PROV has been adopted by a number of workflow systems for encoding the traces of workflow executions. Exploiting these provenance traces is hampered by two main impediments. Firstly, workflow systems extend PROV differently to cater for system-specific constructs. The difference between the adopted PROV extensions yields heterogeneity in the generated provenance traces. This heterogeneity diminishes the value of such traces, *e.g.* when combining and querying provenance traces of different workflow systems. Secondly, the provenance recorded by workflow systems tends to be large, and as such difficult to browse and understand by a human user. In this paper[4], we propose SHARP, a Linked Data approach for harmonizing cross-workflow provenance. The harmonization is performed by chasing tuple-generating and equality-generating dependencies defined for workflow provenance. This results in a provenance graph that can be summarized using domain-specific vocabularies. We experimentally evaluate SHARP i) on publicly available provenance documents and ii) using a real-world omic experiment involving workflow traces generated by the Taverna and Galaxy systems.

Keywords: Reproducibility, Scientific Workflows, Provenance, Prov Constraints

1 Introduction

Reproducibility has recently gained momentum in (computational) sciences as a means for promoting the understanding, transparency and ultimately the reuse of experiments. This is particularly true in life sciences where Next Generation Sequencing (NGS) equipments produce tremendous amounts of omics data, and lead to massive computational analysis (aligning, filtering, etc.). Life scientists urgently need for reproducibility and reuse to avoid duplication of storage and computing efforts.

[4] This paper is an extension of [14], initially published at SeWeBMeDA'17.

Workflows have been used for almost two decades as a means for specifying, enacting and sharing scientific experiments. To tackle reproducibility challenges, major workflow systems have been instrumented to automatically track provenance information. Such information specifies, among other things, the data products (entities) that were used and generated by the operations of the experiments and their derivation paths. Workflow provenance has several applications since it can be utilized for debugging workflows, tracing the lineage of workflow results, as well as understanding the workflow and enabling its reuse and reproducibility [22,18,6,4].

Despite the fact that workflow systems are currently adopting extensions of the PROV recommendation [19], the extensions they adopt use different constructs of PROV. An increasing number of provenance-producing environments adopt semantic web technologies and propose/use extensions of the PROV-O ontology [17]. Because of this, exploiting the provenance traces of multiple workflows, enacted by different workflow systems, is hindered by their heterogeneity.

We present in this paper SHARP, a solution that we investigated for harmonizing and linking the provenance traces produced by different workflow systems.

Specifically, we make the following contributions:

- An approach for interlinking and harmonizing provenance traces recorded by different workflow systems based on PROV inferences.
- An application of provenance harmonization towards Linked Experiment Reports by using domain-specific annotations as in [15].
- An evaluation with public PROV documents and a real-world omic use case.

The paper is organized as follows. Section 2 describes motivations and problem statement. Section 3 presents the harmonization of multiple PROV Graphs and its application towards Linked Experiment Reports. Section 4 and 5 report our implementation and experimental results. Section 6 summarizes related works. Finally, conclusions and future works are outlined in Section 7.

2 Motivations and Problem Statement

Due to costly equipments and massively produced data, DNA sequencing is generally outsourced to third-party facilities. Therefore, one part of the experiments is conducted by the sequencing facility requiring dedicated computing infrastructures, and a second part is conducted by the scientists themselves to analyze and interpret the results based on traditional computing resources. Figure 2.1 illustrates a concrete example of two workflows enacted by different workflow systems, namely Galaxy [2] and Taverna [21].

The first workflow (WF1), in blue in Figure 2.1, is implemented in Galaxy and addresses common DNA data pre-processing. Such workflow takes as input two DNA sequences from two biological samples `s1` and `s2`, represented in green. For each sample, the sequence data is stored in forward[5] (`.R1`) and reverse (`.R2`)

[5] DNA sequencers can decode genomic sequences in both forward and reverse directions which improves the accuracy of alignment to reference genomes.

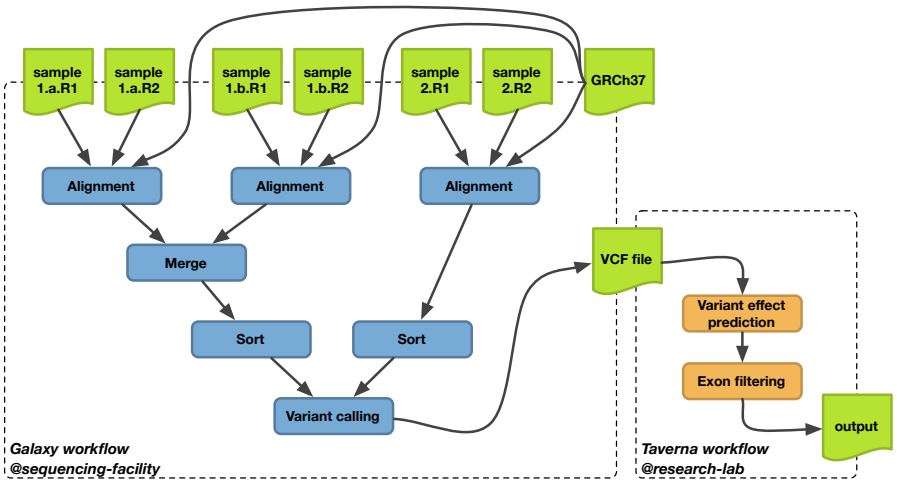


Fig. 2.1: A multi-site genomics workflow, involving Galaxy and Taverna workflow environments.

files. The first sample has been split by the sequencer in two parts, (`.a`) and (`.b`). The very first processing step consists in aligning (`Alignment`[6]) short sequence reads onto a reference human genome (`GRCh37`). Then the two parts `a` and `b` are merged[7] into a single file. Then the aligned reads are sorted[8] prior to genetic variant identification[9] (`Variant Calling`). This primary analysis workflow finally produces a VCF[10] file which lists all known genetics variations compared to the `GCRh37` reference genome.

The second workflow (WF2) is implemented with Taverna, and highly depends on scientific questions. It is generally conducted by life scientists possibly from different research labs and with less computational needs. Such workflow proceeds as follows. It first queries a database of known effects to associate a predicted effect[11] (`Variant effect prediction`). Then all these predictions are filtered to select only those applying to the exon parts of genes (`Exon filtering`). The results obtained by the executions of such workflows allow the scientists to have answers for questions such as Q1 : *"From a set of gene mutations, which are common variants, and which are rare variants ?"*, Q2 : *"Which alignment algorithm was used when predicting these effects ?"*, or Q3: *"A new version of a reference genome is available, which genome was used when predicting these effects ?"*. While Q1 can be answered based on provenance tracking from WF1,

[6] BWA-mem: `http://bio-bwa.sourceforge.net`
[7] PICARD: `https://broadinstitute.github.io/picard/`
[8] SAMtools sort: `http://www.htslib.org`
[9] SAMtools mpileup
[10] Variant Call Format
[11] SnpEff tool: `http://snpeff.sourceforge.net`

Q2 and Q3 need for an overall tracking of provenance at the scale of both WF1 (Galaxy) and WF2 (Taverna) workflows.

While the two workflow environments used in the above experiments (Taverna and Galaxy) track provenance information conforming to the same W3C standardized PROV vocabulary, there are unfortunately impediments that hinder their exploitation. i) The heterogeneity of the provenance languages, despite the fact that they extend the same vocabulary PROV, does not allow the user to issue queries that combine traces recorded by different workflow systems. ii) Heterogeneity aside, the provenance traces of workflow runs tend to be large, and thus cannot be utilized as they are to document the results of the experiment execution. We show how the above issues can be addressed by, i) applying graph saturation techniques and PROV inferences to overcome vocabulary heterogeneity, and ii) summarizing harmonized provenance graphs for life-science experiment reporting purposes.

3 Harmonizing multiple PROV Graphs

Faced with the heterogeneity in the provenance vocabularies, we can use classical data integration approaches such as peer-to-peer data integration or mediator-based data integration [11]. Both options are expensive since they require the specification of schema mappings that often require heavy human inputs. In this paper, we explore a third and cheaper approach that exploits the fact that many of the provenance vocabularies used by workflow systems extend the W3C PROV-O ontology. This means that such vocabularies already come with implicit mappings between the concepts and relationships they used and those of the W3C PROV-O. Of course, not all the concepts and relationships used by individual mappings will be catered for in PROV. Still this solution remains attractive because it does not require any human inputs, since the constraints (mappings) are readily available. We show in this section how the different provenance traces can be harmonized by capitalizing on such constraints.

3.1 Tuple-Generating Dependencies

Central to our approach to harmonizing provenance traces is the saturation operation. Given a possibly disconnected provenance RDF graph $\mathtt{G}$, the saturation process generates a saturated graph $\mathtt{G}^{\infty}$ obtained by repeatedly applying some rules to $\mathtt{G}$ until no new triple can be inferred. We distinguish between two kinds of rules. **OWL entailment rules** includes, among other things, rules for deriving new RDF statements through the transitivity of class and property relationships. **Prov constraints** [8], these are of interest to us as they encode inferences and constraints that need to be satisfied by provenance traces, and can as a such be used for deriving new RDF provenance triples.

In this section, we examine such constraints by identifying those that are of interest when harmonizing the provenance traces of workflow executions, and show (when deemed useful) how they can be translated into SPARQL queries for

saturation purposes. It is worth noting that the W3C Provenance constraint document presents the inferences and constraints assuming a relational-like model with possibly relations of arity greater than 2. We adapt these rules to the context of RDF where properties (relations) are binary. For space limitations, we do not show all the inferences rules that can be implemented in SPARQL, we focus instead on representative ones. We identify three categories of rules with respect to expressiveness (i) rules that contain only universal variables, (ii) rules that contain existential variables, (iii) rules making use of n-array relations (with $n \geqslant 3$). The latter is interesting, since RDF reification is needed to represent such relations. For exemplary rule, we present the rules using tuple-generating dependencies TGDs [1], and then show how we encode it in SPARQL. A TGD is a first order logic formula $\forall \bar{x}\bar{y}\ \phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\ \psi(\bar{y}, \bar{z})$, where $\phi(\bar{x}, \bar{y})$ and $\psi(\bar{y}, \bar{z}))$ are conjunctions of atomic formulas.

Transitivity of alternateOf. Alternate-Of is a binary relation that associates two entities $\mathtt{e_1}$ and $\mathtt{e_2}$ to specify that the two entities present aspects of the same thing. The following rule states that such a relation is transitive, and it can be encoded using a SPARQL construct query, in a straightforward manner.

$$\mathtt{alternateOf}(\mathtt{e_1}, \mathtt{e_2}),\ \mathtt{alternateOf}(\mathtt{e_2}, \mathtt{e_3}) \ \rightarrow\ \mathtt{alternateOf}(\mathtt{e_1}, \mathtt{e_3}).$$

Inference of Usage and Generation from Derivation The following rule states that if an entity $\mathtt{e_2}$ was derived from an entity $\mathtt{e_1}$, then there exists an activity $\mathtt{a}$, such that $\mathtt{a}$ used $\mathtt{e_1}$ and generated $\mathtt{e_2}$.

$$\mathtt{wasDerivedFrom}(\mathtt{e_2}, \mathtt{e_1}) \ \rightarrow\ \exists\ \mathtt{a}\ \mathtt{used}(\mathtt{a}, \mathtt{e_1}),\ \mathtt{wasGeneratedFrom}(\mathtt{e_2}, \mathtt{a}).$$

Notice that unlike the previous rule, the head of the above rule contains an existential variable, namely the activity `a`. To encode such a rule in SPARQL, we make use of blank nodes [12] for existential variables as illustrated below.

```
CONSTRUCT {
    ?e_2 prov:wasGeneratedBy _:blank_node .
    _:blank_node prov:used ?e_1
} WHERE { ?e_2 prov:wasDerivedFrom ?e_1  }
```

Using the Qualification patterns In the previous rule, derivation, usage and generation are represented using binary relationships, which do not pose any problem to be encoded in RDF. Note, however, that PROV-DM allows such relationships to be augmented with optional attributes. For example, usage can be associated with a timestamp specifying the time at which the activity used the entity. The presence of extra optional attributes increases the arity of the relations that can no longer be represented using an RDF property. As a solution, the PROV-O opts for qualification patterns [13] introduced in [12].

The following rule shows how the inference of usage and generation from derivation can be expressed when such relationships are qualified. It can also be encoded using a SPARQL Construct query with blank nodes.

[12] `https://www.w3.org/TR/rdf11-concepts/#dfn-blank-node`

[13] `https://www.w3.org/TR/prov-o/`

$$\texttt{qualifiedDerivation}(\texttt{e}_2, \texttt{d}),\ \texttt{provEntity}(\texttt{d}, \texttt{e}_1) \rightarrow \exists\ \texttt{a},\ \texttt{u},\ \texttt{g}\ \texttt{qualifiedUsage}(\texttt{a}, \texttt{u}),\ \texttt{provEntity}(\texttt{u}, \texttt{e}_1), \texttt{qualifiedGeneration}(\texttt{e}_2, \texttt{g}), \texttt{provActivity}(\texttt{g}, \texttt{a}).$$

Figure 3.1 presents inferred statements in dashed arrows resulting from the application of this rule.

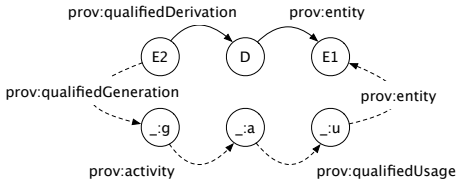


Fig. 3.1: Inferred qualified usage and generation relationships.

3.2 Equality-Generating Dependencies

As well as the tuple-generating dependencies, we need to consider equality-generating dependencies (EGDs), which are induced by uniqueness constraints. An EGD is a first order formula: $\forall \bar{\texttt{x}} \phi(\bar{\texttt{x}}) \rightarrow (\texttt{x}_1 = \texttt{x}_2)$, where $\phi(\bar{\texttt{x}})$ is a conjunction of atomic formulas, and $\texttt{x}_1$ and $\texttt{x}_2$ are among the variables in $\bar{\texttt{x}}$. We give below an example of an EGD, that is implied by the uniqueness of the generation that associates a given activity a with a given entity e.

$$\texttt{wasGeneratedBy}(\texttt{gen}_1, \texttt{e}, \texttt{a}, \texttt{attrs}_1),\ \texttt{wasGeneratedBy}(\texttt{gen}_2, \texttt{e}, \texttt{a}, \texttt{attrs}_2) \rightarrow (\texttt{gen}_1 = \texttt{gen}_2)$$

Having defined an example EGD, we need to specify what it means to apply it (or chase it [13]) when we are dealing with RDF data. The application of an EGD has three possible outcomes. To illustrate them, we will work on the above example EGD. Typically, the generations $\texttt{gen}_1$ and $\texttt{gen}_2$ will be represented by two RDF resources. We distinguish the following cases:

(i) $\texttt{gen}_1$ **is a non blank RDF resource and** $\texttt{gen}_2$ **is a blank node.** In this case, we add to $\texttt{gen}_1$ the properties that are associated with the blank node $\texttt{gen}_2$, and remove $\texttt{gen}_2$. (ii) $\texttt{gen}_1$ **and** $\texttt{gen}_2$ **are two blank nodes.** In this case, we create a single blank node gen to which we associate the properties obtained by unionizing the properties of $\texttt{gen}_1$ and $\texttt{gen}_2$, and we remove the two initial blank nodes. (iii) $\texttt{gen}_1$ **and** $\texttt{gen}_2$ **are non blank nodes that are different.** In this case, the application of the EGD (as well as the whole saturation) fails. In general, we would not have this case, if the initial workflows runs that we use as input are valid (ie., they respect the constraints defined in the W3C Prov Constraint recommendation [8].

Algorithm 1: EGD pseudo-code for merging blank nodes produced by PROV inference rules with existential variables.

Input : G' : the provenance graph resulting from the application of TGD on G
Output: G'': the provenance graph with substituted blank nodes, when possible.

```
1  begin
2      G'' ← G'
3      substitutions ← new List < Pair < Node, Node >> ()
4      repeat
5          S ← findSubstitutions(G')
6          foreach (s ∈ S) do
7              source ← s[0]
8              target ← s[1]
9              foreach (in ∈ G'.listStatements(∗, ∗, source)) do
10                 G'' ← G''.add(in.getSubject(), in.getPredicate(), target)
11                 G'' ← G''.del(in)
12             foreach (out ∈ G'.listStatements(source, ∗, ∗)) do
13                 G'' ← G''.add(target, out.getPredicate(), out.getObject())
14                 G'' ← G''.del(out)
15     until (S.size() = 0)
```

To select the candidate substitutions (line 5 of Algorithm 1), we express the graph patterns illustrated in the previous cases 1 and 2 as a SPARQL query. This query retrieves candidate substitutions as blank nodes coupled to their substitute, *i.e.*, another blank node or a URI.

For each of the found substitution (line 6), we merge the incoming and outgoing relations between the source node and the target node. This operation is done in two steps. First, we navigate through the incoming relations of the source node (line 9), we copy them as incoming relations of the target node (line 10), and finally remove them from the source node (line 11). Second, we repeat this operation for the outgoing relations (lines 12 to 14). We repeat this process until we can't find any candidate substitutions.

3.3 Full provenance harmonization process

The full provenance harmonization workflow is sketched in Figure 3.2.

❶ **Multi-provenance linking.** This process starts by first linking the traces of the different workflow runs. Typically, the outputs produced by a run of a given workflow are used to feed the execution of a run of another workflow as depicted in Figure 2.1.

The main idea consists in providing an *owl:sameAs* property between the PROV entities associated with the same physical files. The production of *owl: sameAs* can be automated as follows : i) generate a fingerprint of the files (SHA-512 is one of the recommended hashing functions), ii) produce the PROV annotation associated the fingerprint to the PROV entities, iii) generate, through a SPARQL CONSTRUCT query, the *owl:sameAs* relationships when fingerprints are matched. When applied to our motivating example (Figure 2.1), the

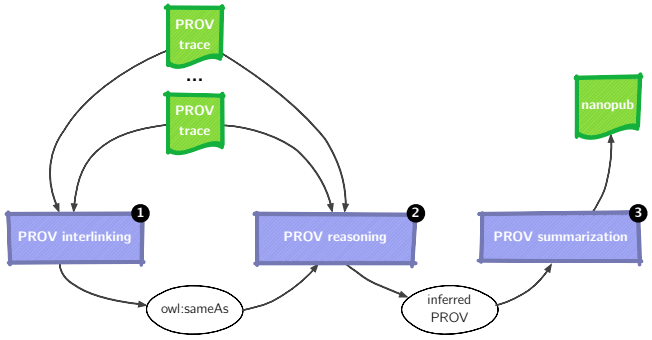


Fig. 3.2: From multiple PROV traces to linked experiment summaries.

PROV entity annotating the $VCFFile$ produced by the Galaxy workflow becomes equivalent to the one as input of Taverna workflow. A PROV example associating a file name and its fingerprint is reported below:

```
<http://fr.symetric#c583bef6-de69-4caa-bc3a-00000000>
        a               prov:Entity ;
        rdfs:label      "my-variants.vcf"^^xsd:String ;
        crypto:sha512   "1d305986330304378f82b938d776ea0be48eda8210f7af6c
        152e8562cf6393b2f5edd452c22ef6fe8c729cb01eb3687ac35f1c5e57ddefc4
        6276e9c60409276a"^^xsd:String .
```

The following SPARQL Construct query can be used to produce `owl:sameAs` relationships :

```
CONSTRUCT { ?x owl:sameAs ?y }
WHERE {
        ?x a prov:Entity .
        ?x crypto:sha512 ?x_sha512 .
        ?y a prov:Entity .
        ?y crypto:sha512 ?y_sha512 .
        FILTER( ?x_sha512 = ?y_sha512 ) }
```

❷ **Multi-provenance reasoning.** Once the traces of the workflow runs have been linked, we saturate the graph obtained using OWL entailment rules. This operation can be performed using an existing OWL reasoner (e.g., [7,16]). We then start by repeatedly applying the TGDs and EGDs derived from the W3C PROV constraint document, as illustrated in section 3.1 and 3.2. The harmonization process terminates when we can no longer apply any existing TGD or EGD. This harmonization process raises the question as to whether such process will terminate. The answer is affirmative. Indeed, it has been shown in the W3C PROV Constraint document that the constraints are weakly acyclic, which guarantees the termination of the chasing process in polynomial time (see Fagin *et al.* [13] for more details).

❸ **Harmonized provenance summarization.** The previously described reasoning step may lead to intractable provenance graphs from a human perspective,

both in terms of size and lack of domain-specificity. We propose in this last step to make sense of the harmonized provenance through domain-specific provenance summaries. This application is described in the following section.

3.4 Application of provenance harmonization: domain-specific experiment reports

In this section we propose to exploit harmonized provenance graphs by transforming them into *Linked Experiment Reports*. These reports are no longer machine-only-oriented and benefit from a humanly tractable size, and domain-specific concepts.

Domain-specific vocabularies. *Workflow annotations.* P-Plan[14] is an ontology aimed at representing the plans followed during a computational experiment. *Plans* can be atomic or composite and are a made by a sequence of processing *Steps*. Each *Step* represents an executable activity, and involves input and output *Variables*. P-Plan fits well in the context of multi-site workflows since it allows to work at the scale of a site-specific workflow as well as at the scale of the global workflow.

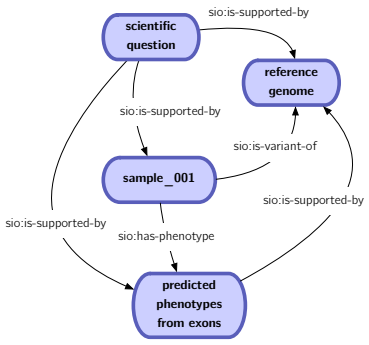
Domain-specific concepts and relations. To capture knowledge associated to the data processing steps, we rely on EDAM[15] which is actively developed in the context of the Bio.Tools bioinformatics registry. However these annotations on processing tools do not capture the scientific context in which a workflow takes place. SIO[16], the Semantic science Integrated Ontology, has been proposed as a comprehensive and consistent knowledge representation framework to model and exchange physical, informational and processual entities. Since SIO has been initially focusing on Life Sciences, and is reused in several Linked Data repositories, it provides a way to link the data routinely produced by PROV-enabled workflow environment to major linked open data repositories, such as Bio2RDF.

NanoPublications[17] are minimal sets of information to publish data as citable artifacts while taking into account the attribution and authorship. NanoPublications provide named graphs mechanisms to link *Assertion*, *Provenance*, and *Publishing* statements. In the remainder of this section, we show how fine-grained and machine-oriented provenance graphs can be summarized into NanoPublications.

Linked Experiment Reports Based on harmonized multi-provenance graphs, we show how to produce NanoPublications as exchangeable and citeable scientific experiment reports. Figure 3.3 drafts how data artifacts and scientific context can be related to each other into a NanoPublication, for the motivating scenario

[14] `http://purl.org/net/p-plan`
[15] `http://edamontology.org`
[16] `http://sio.semanticscience.org`
[17] `http://nanopub.org`

introduced in section 2. For the sake of simplicity we omitted the definition of namespaces, and we used the labels of SIO predicates instead of their identifiers.




```
:head {
    ex:pub1 a np:Nanopublication .
    ex:pub1 np:hasAssertion :assertion1 ;
        np:hasAssertion :assertion2 .
    ex:pub1 np:hasProvenance :provenance .
    ex:pub1 np:hasPublicationInfo :pubInfo .}
:assertion1 {
    ex:question a sio:Question ;
        sio:has-value "What are the effects of SNPs
        located in exons for study-Y samples"  ;
        sio:is-supported-by ex:referenceGenome ;
        sio:is-supported-by ex:sample_001 ;
        sio:is-supported-by ex:annotatedVariants .}
:assertion2 {
    ex:referenceGenome a sio:Genome .
    ex:sample_001 a sio:Sample ;
        sio:is-variant-of ex:referenceGenome ;
        sio:has-phenotype ex:annotatedVariants .}
:provenance { :assertion2 prov:wasDerivedFrom
    :harmonizedProvBundle .}
:pubInfo { ex:pub1 prov:wasAttributedTo ex:MyLab .}
```

Fig. 3.3: Graphical and RDF representation of an experiment report, providing context and linking the most relevant multi-site workflow artifacts to domain specific statements.

To produce this NanoPublication, we identify a data lineage path in multiple PROV graphs, beforehand harmonized (as proposed in section 3). Since we identified the *prov:wasInfluencedBy* as the most commonly inferred lineage relationship, we search for all connected data entities through this relationship. Then, when connected data entities are identified, we extract the relevant ones so that they can be later on incorporated and annotated through new statements in the NanoPublication. The following SPARQL query illustrates how `:assertion2` can be assembled from a matched path in harmonized provenance graphs. The key point consists in relying on SPARQL property path expressions `(prov:wasInfluencedBy)+` to identify all paths connecting data artifacts composed by one or more occurrences of the *prov:wasInfluencedBy* predicate. Such SPARQL queries could be programmatically generated based on P-Plan templates as it has been proposed in our previous work [15].

```
CONSTRUCT {
    GRAPH :assertion {
        ?ref_genome a sio:Genome .
        ?sample a sio:Sample ;
                sio:is-variant-of ?ref_genome ;
                sio:has-phenotype ?out .
        ?out rdfs:label ?out_label .
        ?out sio:is-supported-by ?ref_genome . }
} WHERE {
   ?sample rdfs:label ?sample_label.
   FILTER (contains(lcase(str(?sample_label)), lcase("fastq"))) .
   ?ref_genome rdfs:label ?ref_genome_label.
   FILTER (contains(lcase(str(?ref_genome_label)), lcase("GRCh"))) .
```

```
?out ( prov:wasInfluencedBy )+ ?sample
?out tavernaprov:content ?out_label .
FILTER (contains(lcase(str(?out_label)), lcase("exons"))) . }
```

4 Implementation

Although Taverna allows to export PROV traces, this is not yet the case for the Galaxy workbench[18]. We thus developed an open-source provenance capture tool[19] for Galaxy. Users provide the URL of their Galaxy workflow portal, and their private API key. Then, the tool communicates with the Galaxy REST API to produce PROV RDF triples. We implemented the full PROV harmonization process (Fig. 3.2) in the sharp-prov-toolbox[20]. This open-source tool has been implemented in Java and is supported by Jena[21] for RDF data management and reasoning. PROV Constraints[22] inference rules have been implemented in the Jena syntax[23]. HTML and JavaScript code templates have been used to generate harmonized provenance visualization. Figure 4.1 shows the resulting data lineage graph associated with the two workflow traces of our motivating use case (Figure 2.1). While the left part of the graphs represents the Galaxy workflow invocation, the right part represents the Taverna one.

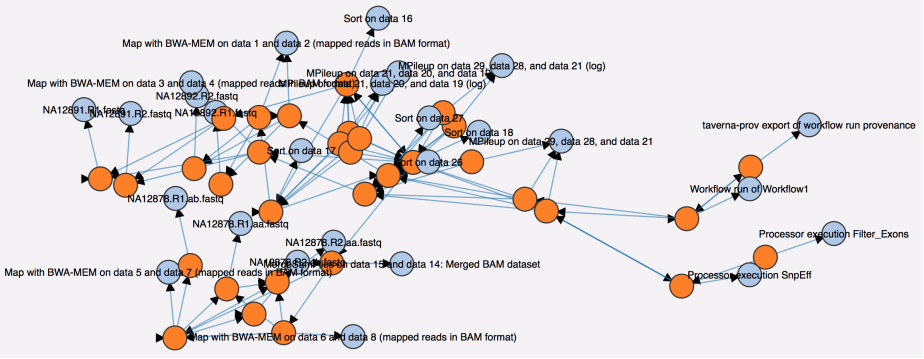


Fig. 4.1: *prov:wasInfluencedBy* properties between Galaxy and Taverna.

[18] https://usegalaxy.org
[19] galaxy-PROV: https://github.com/albangaignard/galaxy-PROV
[20] sharp-prov-toolbox: https://github.com/albangaignard/sharp-prov-toolbox
[21] Jena: https://jena.apache.org
[22] https://www.w3.org/TR/prov-constraints/
[23] https://github.com/albangaignard/sharp-prov-toolbox/blob/master/SharpProvToolbox/src/main/resources/provRules_all.jena

5 Experimental results and discussion

As a first evaluation, we ran two experiments. The first one evaluates the harmonization process at large scale. In a second experiment, we evaluated the ability of the system to answer the domain-specific questions of our motivating scenario.

5.1 Harmonization of heterogeneous PROV traces at large scale

In this experiment, we used provenance documents from ProvStore[24]. We selected the 369 public documents of 2016. These documents have different sizes from 1 to 58572 triples and use different PROV concepts and relations. We ran the provenance harmonization process as described in this paper on a classical desktop computer (4-cores CPU, 16GB of memory). From the initial 217165 PROV triples, it took 38mn to infer 1291549 triples. Each provenance document has been uploaded as a named graph to a Jena Fuseki endpoint. The two histograms of Figure 5.1 show the number of named graphs in which PROV predicates are present. We filtered the predicates to show only predicates using the PROV prefix. Figure 5.1 shows that we have been able to harmonize (right histogram, in orange) the provenance documents since we increase the number of named graphs in which PROV predicates are inferred. Specifically, we have been able to infer new *influence* relations in 318 provenance documents.

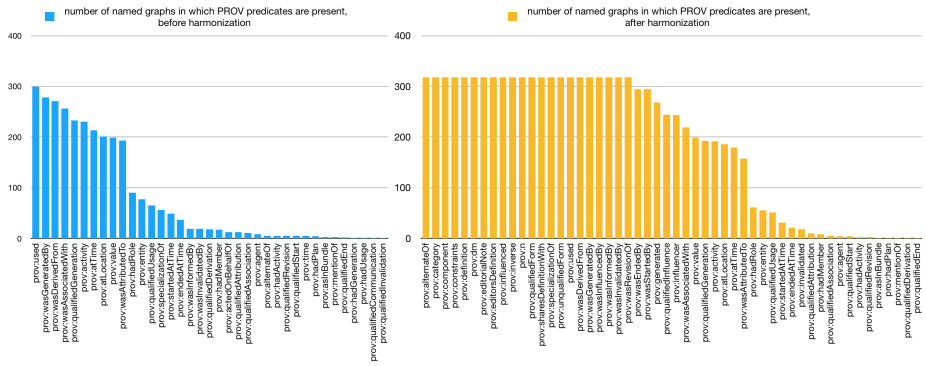


Fig. 5.1: Distribution of asserted (blue) and inferred (orange) PROV predicates in the public ProvStore documents for year 2016, before and after the proposed harmonization process.

5.2 Usage of semi-automatically produced NanoPublications

We run the multi-site experiment of section 2 using Galaxy and Taverna workflow management systems. The Galaxy workflow has been designed in the context of

[24] `https://provenance.ecs.soton.ac.uk/store/`

the SyMeTRIC systems medicine project, and was run on the production Galaxy instance[25] of the BiRD bioinformatics infrastructure. The Taverna workflow was run on a desktop computer. Provenance graphs were produced by the Taverna built-in PROV feature, and by a Galaxy dedicated provenance capture tool[26], based on the Galaxy API, the later transforms a user history of actions into PROV RDF triples.

Table 1 presents a sorted count of the top-ten predicates in i) the Galaxy and Taverna provenance traces without harmonization, ii) these provenance traces after the first iteration of the harmonization process:

Galaxy PROV		*Taverna PROV*		*Harmonized PROV++*	
predicates	**counts**	**predicates**	**counts**	**predicates**	**counts**
prov:wasDerivedFrom	118	rdf:type	54	owl:differentFrom	3617
rdf:type	76	rdfs:label	13	rdf:type	958
rdfs:label	62	prov:atTime	8	prov:wasInfluencedBy	515
prov:used	61	wfprov:descByParameter	6	prov:influenced	291
prov:wasAttributedTo	34	rdfs:comment	6	rdfs:seeAlso	268
prov:wasGeneratedBy	33	prov:hadRole	6	rdfs:subClassOf	223
prov:endedAtTime	26	prov:activity	5	owl:disjointWith	218
prov:startedAtTime	26	purl:hasPart	4	rdfs:range	208
prov:wasAssociatedWith	26	prov:agent	4	rdfs:domain	199
prov:generatedAtTime	1	prov:endedAtTime	4	prov:wasGeneratedBy	172
all	463	*all*	177	*all*	8654

Table 1: Most prominent predicates when considering the initial two PROV graphs and their harmonization (*PROV++*)

We executed the summarization query proposed in section 3.4 on the harmonized provenance graph. The resulting NanoPublication (*assertion* named graph) represents the input DNA sequences aligned to the GRCh37 human reference genome through an *sio:is-variant-of* predicate. It also links the annotated variants (Taverna WF output) with the preprossessed DNA sequences (Galaxy WF inputs). Related to the Q3 life-science question highlighted in section 2, this NanoPublication can be queried to retrieve for instance the reference genome used to select and annotate the resulting genetic variants.

6 Related Works

Data integration [11] and summarization [3] have been largely studied in different research domains. Our objective is not to invent yet another technique for integrating and/or summarizing data. Instead, we show how provenance constraint rules, domain annotations, and Semantic Web techniques can be combined to harmonize and summarize provenance data into linked experiment reports.

Several proposals tackle scientific reproducibility [27]. For example, Reprozip [9] captures operating system events that are then utilized to generate a workflow

[25] `https://galaxy-bird.univ-nantes.fr/galaxy/`

[26] `https://github.com/albangaignard/sharp-prov-toolbox`

[27] `http://www.refinery-platform.org`

illustrating the events that happened and their sequences. While valuable, such proposals neither address the harmonization of multi-systems and heterogeneous provenance traces nor machine- and human-tractable experiment reports, as proposed in SHARP.

Datanode ontology [10] proposes to harmonize data by describing relationships between data artifacts. Datanode allows to present in a simple way dataflows that focus on the fundamental relationships that exist between original, intermediary, and final datasets. Contrary to Datanode, SHARP uses existing PROV vocabularies and constraints to harmonize provenance traces, thereby reducing harmonization efforts.

LabelFlow [5] proposes a semi-automated approach for labeling data artifacts generated from workflow runs. Compared to LabelFlow, SHARP uses existing PROV ontology and Semantic Web technology to harmonize dataflows. Moreover, *LabelFlow* is confined to single workflows, whereas SHARP targets a collection of workflow runs that are produced by different workflow systems.

In previous work [15], we proposed *PoeM* to produce linked in silico experiment reports based on workflow runs. As SHARP, *PoeM* leverages Semantic Web technologies and reference vocabularies (PROV-O, P-Plan) to generate provenance mining rules and finally assemble linked scientific experiment reports (Micropublications, Experimental Factor Ontology). SHARP goes steps forward by proposing the harmonization of multi-systems provenance traces.

7 Conclusions

In this paper, we presented SHARP, a Linked Data approach for harmonizing cross-workflow provenance. The resulting harmonized provenance graph can be exploited to run cross-workflow queries and to produce provenance summaries, targeting human-oriented interpretation and sharing. Our ongoing work includes deploying SHARP to be used by scientists to process their provenance traces or those associated with provenance repositories, such as ProvStore. For now, we work on multi-site provenance graphs with centralized inferences. Another exciting research direction would be to consider low-cost highly decentralized infrastructure for publishing NanoPublication as proposed in [20].

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
2. E. Afgan, D. Baker, van den Beek, et al. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Research*, 44(W1):W3–W10, 2016.
3. C. C. Aggarwal and H. Wang. Graph data management and mining: A survey of algorithms and applications. In *Managing and Mining Graph Data*, pages 13–68. Springer, 2010.

4. P. Alper, K. Belhajjame, C. A. Goble, and P. Karagoz. Enhancing and abstracting scientific workflow provenance for data publishing. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, pages 313–318. ACM, 2013.
5. P. Alper, K. Belhajjame, C. A. Goble, and P. Karagoz. Labelflow: Exploiting workflow provenance to surface scientific data provenance. In *Provenance and Annotation of Data and Processes - 5th International Provenance and Annotation Workshop, IPAW 2014, Cologne, Germany, June 9-13, 2014. Revised Selected Papers*, pages 84–96, 2014.
6. I. Altintas, O. Barney, and E. Jaeger-Frank. Provenance collection support in the kepler scientific workflow system. In *Provenance and annotation of data*, pages 118–132. Springer, 2006.
7. J. J. Carroll, I. Dickinson, et al. Jena: implementing the semantic web recommendations. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 74–83. ACM, 2004.
8. J. Cheney, P. Missier, and L. Moreau. Constraints of the provenance data model. Technical report, 2012.
9. F. Chirigati, D. Shasha, and J. Freire. Reprozip: Using provenance to support computational reproducibility. In *5th USENIX Workshop on the Theory and Practice of Provenance*, Berkeley, CA, 2013.
10. E. Daga, M. d'Aquin, et al. Describing semantic web applications through relations between data nodes, 2014.
11. A. Doan, A. Halevy, and Z. Ives. *Principles of Data Integration*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2012.
12. L. Dodds and I. Davis. *Linked Data Patterns: A pattern catalogue for modelling, publishing, and consuming Linked Data*. May 2012.
13. R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: semantics and query answering. *Theoretical Computer Science*, 336(1):89–124, 2005.
14. A. Gaignard, K. Belhajjame, and H. Skaf-Molli. Sharp: Harmonizing cross-workflow provenance. In *SeWeBMeDA Workshop on semantic web solutions for large-scale biomedical data analytics*, 2016.
15. A. Gaignard, H. Skaf-Molli, and A. Bihouée. From scientific workflow patterns to 5-star linked open data. In *8th USENIX Workshop on the Theory and Practice of Provenance*, 2016.
16. Jena. Reasoners and rule engines: Jena inference support. *The Apache Software Foundation*, 2013.
17. T. Lebo, S. Sahoo, D. McGuinness, et al. Prov-o: The prov ontology. *W3C Recommendation*, 30, 2013.
18. S. Miles, P. Groth, M. Branco, and L. Moreau. The requirements of using provenance in e-science experiments. *Journal of Grid Computing*, 5(1):1–25, 2007.
19. P. Missier, K. Belhajjame, and J. Cheney. The w3c prov family of specifications for modelling provenance metadata. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 773–776. ACM, 2013.
20. K. T, C. C, K. M, et al. Decentralized provenance-aware publishing with nanopublications. *PeerJ Computer Science 2:e78 https://doi.org/10.7717/peerj-cs.78*.
21. K. Wolstencroft, R. Haines, D. Fellows, et al. The taverna workflow suite: designing and executing workflows of web services on the desktop, web or in the cloud. *Nucleic Acids Research*, 41(Webserver-Issue):557–561, 2013.
22. J. Zhao, C. Wroe, C. Goble, et al. Using semantic web technologies for representing e-science provenance. In *The Semantic Web–ISWC 2004*, pages 92–106. Springer, 2004.