SHARP: Harmonizing cross-workflow Provenance

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

[1] Nantes Academic Hospital, France
alban.gaignard@univ-nantes.fr
[2] LAMSADE – Paris-Dauphine University, France
kbelhajj@googlemail.com
[3] LINA – Nantes University, 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, 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 the effectiveness of SHARP 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 scientific experiments. This is particularly true in the life sciences where Next Generation Sequencing (NGS) equipments produce tremendous amounts of omics data, and lead to massive computational analysis (aligning, comparing, filtering, etc.). Life scientists urgently need for reproducibility and reuse to avoid duplication of storage and computing efforts.

Pivotal to reproducibility is provenance [11], which documents the experiment, including information about the activities that were conducted during the experiment, the agents that were involved, the resources and programs that

were utilized as well as the data artifacts that were used and generated. Several researchers have investigated the use of provenance as a means for tracing pack the execution of experiment (see e.g., [23,19,6,4]). We note however that experiments may involve multiple scientists, each of them is responsible for conducting and analyzing the execution of part of the overall experiment, using his/her favorite data analysis tool (workflow system, programming or scripting language, etc.), which may be different from those used by the rest of the team. This is particularly the case for interdisciplinary projects involving scientists with different backgrounds and expertise. In order to exploit the provenance generated by the different data analysis tools utilized within the scope of an experiment, there is therefore the need for harmonizing and interlinking the provenance traces such tool recorded and generated. The adoption of the W3C PROV recommendations [20] (in particular the PROV-O ontology [18] given increasing number of provenance-producing environments adopting semantic web technologies) by a number of data analysis tools has to a certain extent lessen the severity of the provenance harmonization problem. Yet, the fact that such environments use PROV extensions that extend differently PROV, means that there is a need for aligning the provenance traces generated by those tools. Moreover, the provenance graphs generated by those environments need to be interlinked by identifying the entities that refer to the same real world entity.

Interlinking and harmonizing provenance data is essential to deliver a global account of what happened during scientific experiments. It is, however, by no mean sufficient for promoting the understanding and re-usability of the experiment and its associated results. Indeed, the provenance graph generated are often large and contain low level and cumbersome information that is targeted for the consumption of machines. This calls for abstraction mechanisms for providing a human user with a global view on what happens in the experiment, by deriving from the raw provenance information, high level and succinct information that helps users in understanding the experiment and the results of its execution in its entirety. In this paper, we propose SHARP that addresses the above issues. We propose 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 real world omic use case illustrating the feasibility of SHARP.

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

2 Motivations and Problem Statement

Due to costly sequencing equipment and massively produced data, DNA sequencing is generally outsourced to third-party facilities. Therefore, part of the scientific experiment is conducted by the sequencing facility which requires dedicated high throughput computing infrastructures, and a second part conducted by the scientists themselves to analyze and interpret the results of sequencing using traditional computing resources. Figure 2.1 illustrates a concrete example of such experiment, which is composed of two workflows enacted by different workflow systems, namely Galaxy [2] and Taverna [22].

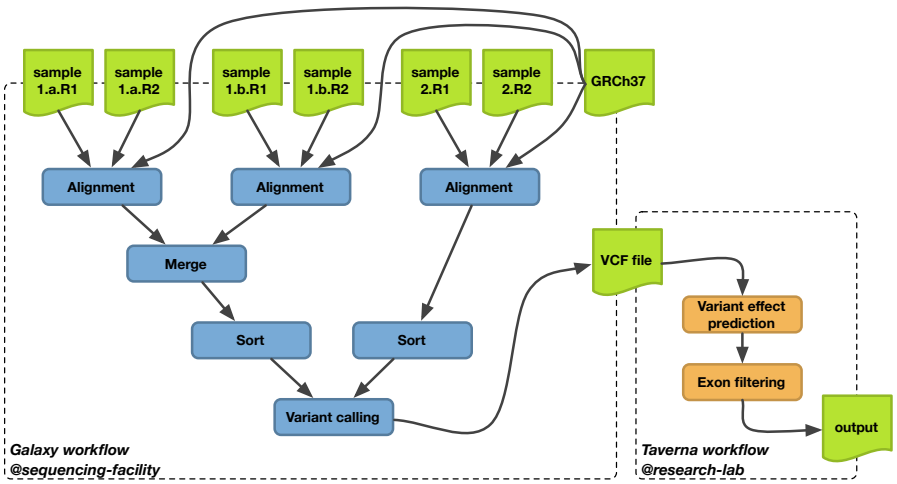

Fig. 2.1: A multi-site genomics workflow, involving Galaxy and Taverna workflow environments.

The first workflow (WF1), in blue in Figure 2.1, is implemented in Galaxy and addresses DNA data pre-processing which is loosely coupled to scientific hypothesis. 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[4] (`.R1`) and reverse (`.R2`) 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`[5]) short sequence reads onto a reference human genome (`GRCh37`). Then the two parts `a` and `b` are merged[6] into a single file. Then

[4] DNA sequencers can decode genomic sequences in both forward and reverse directions which improves the accuracy of alignment to reference genomes.
[5] BWA-mem: `http://bio-bwa.sourceforge.net`
[6] PICARD: `https://broadinstitute.github.io/picard/`

the aligned reads are sorted[7] prior to genetic variant identification[8] (`Variant Calling`). This primary analysis workflow finally produces a VCF[9] file which list 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 query a database of known effects to associate a predicted effect[10] (`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, 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, which can be valuable, there are unfortunately impediments that hinder their exploitation. i)- The heterogeneity of the provenance languages used to encode workflow runs, despite the fact that they extend the same vocabulary PROV, does not allow the user to issue queries that use and combine traces recorded by different workflow languages. 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 multi-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 [12] 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

[7] SAMtools sort: `http://www.htslib.org`
[8] SAMtools mpileup
[9] Variant Call Format
[10] SnpEff tool: `http://snpeff.sourceforge.net`

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 provenance traces that are encoded using different PROV extensions 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 [11] 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
}
```

Inference of Usage and Generation from Derivation 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 [12] introduced in [13]. To illustrate this, Figure 3.1 shows how a qualified usage can be encoded in RDF.

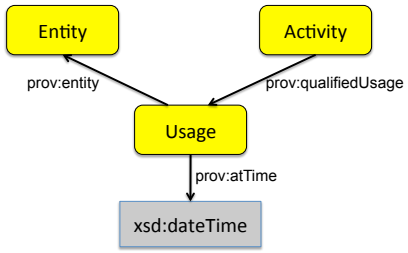

Fig. 3.1: Example of a qualified relationship.

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.

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

Figure 3.2 presents inferred statements in dashed arrows resulting from the application of this rule.

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

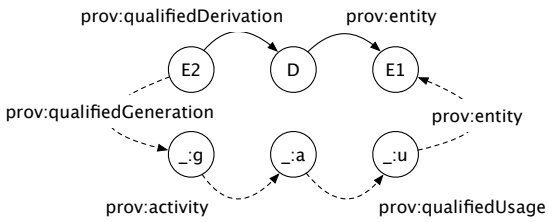

Fig. 3.2: Inferred qualified usage and generation relationships.

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: $\forall \bar{\mathtt{x}} \phi(\bar{\mathtt{x}}) \rightarrow (\mathtt{x}_1 = \mathtt{x}_2)$, where $\phi(\bar{\mathtt{x}})$ is a conjunction of atomic formulas, and $\mathtt{x}_1$ and $\mathtt{x}_2$ are among the variables in $\bar{\mathtt{x}}$. We give below an examples of an EGD, that is implied by the uniqueness of the generation that associates a given activity $\mathtt{a}$ with a given entity $\mathtt{e}$.

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

Having defined an example EGD, we need to specify what it means to apply it (or chase it [14]) 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 $\mathtt{gen}_1$ and $\mathtt{gen}_2$ will be represented by two RDF resources. We distinguish the following cases:

(i) $\mathtt{gen}_1$ **is a non blank RDF resource and** $\mathtt{gen}_2$ **is a blank node.** In this case, we add to $\mathtt{gen}_1$ the properties that are associated with the blank node $\mathtt{gen}_2$, and remove $\mathtt{gen}_2$. (ii) $\mathtt{gen}_1$ **and** $\mathtt{gen}_2$ **are two blank nodes.** In this case, we create a single blank node $\mathtt{gen}$ to which we associate the properties obtained by unionizing the properties of $\mathtt{gen}_1$ and $\mathtt{gen}_2$, and we remove the two initial blank nodes. (iii) $\mathtt{gen}_1$ **and** $\mathtt{gen}_2$ **are non black 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].

To select the candidate substitutions (line 5 of the algorithm), 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

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)
```

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

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 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 ) }
```

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., [7,17]). 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.* [14] for more details).

3.4 Application of provenance harmonization: domain-specific experiment reports

In this section we propose to exploit previously harmonized provenance graphs by transforming them into *Linked Experiment Reports*. These reports are no more machine-only-oriented and benefit from a humanly tractable size, and domain-specific concepts. Several ontologies and controlled vocabularies have been proposed to capture and organize knowledge associated to *in silico* experiments.

Domain-specific vocabularies. *Workflow annotations.* P-Plan[13] 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[14] which is actively developed, in the context of the Bio.Tools registry, and which organizes common terms used in the field of bioinformatics. However these annotations on processing tools do not capture the scientific context in which a workflow takes place. SIO[15], 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

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

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[16] 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 for the motivating scenario introduced in section 2.

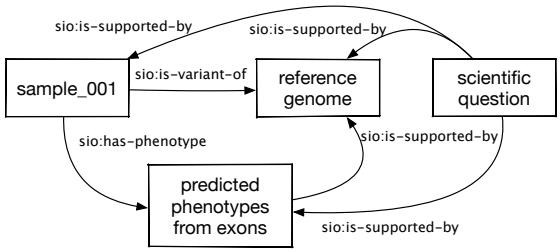

Fig. 3.3: Expected experiment report, linking the most relevant multi-site workflow artifacts with domain specific statements, and scientific context.

The expected Linked Experiment Report would be a NanoPublication as follows. 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 .
```

[16] `http://nanopub.org`

```
   ex:annotatedVariants sio:is-supported-by ex:referenceGenome . }
:provenance { :assertion2 prov:wasDerivedFrom :harmonizedProvBundle .}
:pubInfo { ex:pub1 prov:wasAttributedTo ex:MyLab . }
```

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 Experimental results and discussion

As a first evaluation, we ran two experiments. The first one evaluates the performance of harmonization in terms of execution time, number and nature of inferred relations. In a second experiment, we evaluated the ability of the system to answer the domain-specific questions of our motivating scenario.

4.1 Harmonization of heterogeneous PROV traces

In this experiment, we used provenance document of ProvStore[17]. Specifically, we selected three documents, namely P_A (ID 113207), P_B (ID 113206), and P_C (ID 113263). These documents have different sizes from 10 to 666 triples and use different concepts and relations of PROV. We ran the provenance harmonization process as described in this paper using such documents on a classical desktop computer (4-cores CPU, 16GB of memory). We computed the mean time and

	size	blank nodes	wDF pred.	wIB pred.	mean time (ms)
P_A	[10,2786]	[0,2]	[0,1]	[0,7]	4835 ± 343
P_B	[109,3211]	[0,4]	[10,11]	[0,58]	4759 ± 71
P_C	[666,5689]	[1,64]	[17,18]	[0,231]	5304 ± 176

Table 1: [*before*,*after*] metrics characterizing the impact of the provenance harmonization process. wDF refers to wasDerivedFrom properties and wIB refers to wasInfluencedBy.

standard deviation based on five executions of the harmonization, as well as the size of the provenance graph before and after the harmonization.

The processing time of the OWL entailments, TGDs, and EGDs provenance harmonization process is near to 5 seconds as shown in Table 1. This is negligible in the context of scientific workflows, which generally rely on possibly long batch job submissions. With respect to the inferred predicates, Table 1 also shows that the number of *wasInfluencedBy* (wIB) is important. In spite of its loose semantics, these inferred statements could be helpful for tracing data lineage in provenance graphs. Even if not present in the original PROV graph, SHARP was able to produce these common data lineage relations. We can also note that the harmonization process does not allow to infer *wasDerivedFrom* (wDF) relations. By design, the PROV inference regime does not allow the inference of new *wasDerivedFrom* relations, which means that a particular attention should be paid to initially capture this provenance relation.

4.2 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 the SyMeTRIC systems medicine project, and was run on the production Galaxy instance[18] 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[19], based on the Galaxy API, the later transforms a user history of actions into PROV RDF triples.

Table 2 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:

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 prepossessed DNA sequences (Galaxy

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

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

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

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

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.

5 Related Works

Data harmonization (integration) [12] 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.

There have been several proposals and tools that tackle scientific reproducibility [20]. For example, Reprozip [9] captures operating system events that are then utilized to generate a workflow illustrating the events that happened and their sequences. While valuable, such proposals neither address the harmonization of provenance traces recorded by different analysis tools that utilize different PROV extension 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 connect and harmonizes the dataflows. Moreover, *LabelFlow* is confined to single workflows, whereas SHARP

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

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 provenance traces recorded by different workflow systems.

6 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 [21].

References

1. Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
2. Enis Afgan, Dannon 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. Charu C. Aggarwal and Haixun Wang. Graph data management and mining: A survey of algorithms and applications. In *Managing and Mining Graph Data*, pages 13–68. Springer, 2010.
4. Pinar Alper, Khalid Belhajjame, Carole A Goble, and Pinar 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. Pinar Alper, Khalid Belhajjame, Carole A. Goble, and Pinar 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. Ilkay Altintas, Oscar Barney, and Efrat Jaeger-Frank. Provenance collection support in the kepler scientific workflow system. In *Provenance and annotation of data*, pages 118–132. Springer, 2006.
7. Jeremy J Carroll, Ian 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. James Cheney, Paolo Missier, and Luc Moreau. Constraints of the provenance data model. Technical report, 2012.
9. Fernando Chirigati, Dennis Shasha, and Juliana Freire. Reprozip: Using provenance to support computational reproducibility. In *5th USENIX Workshop on the Theory and Practice of Provenance*, Berkeley, CA, 2013.
10. Enrico Daga, Mathieu d'Aquin, et al. Describing semantic web applications through relations between data nodes, 2014.
11. Susan B. Davidson and Juliana Freire. Provenance and scientific workflows: challenges and opportunities. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1345–1350. ACM, 2008.
12. AnHai Doan, Alon Halevy, and Zachary Ives. *Principles of Data Integration*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2012.
13. Leigh Dodds and Ian Davis. *Linked Data Patterns: A pattern catalogue for modelling, publishing, and consuming Linked Data*. May 2012.
14. Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theoretical Computer Science*, 336(1):89–124, 2005.
15. Alban Gaignard, Hala Skaf-Molli, and Audrey Bihouée. From scientific workflow patterns to 5-star linked open data. In *8th USENIX Workshop on the Theory and Practice of Provenance*, 2016.
16. Robert Isele and Christian Bizer. Active learning of expressive linkage rules using genetic programming. *Web Semantics: Science, Services and Agents on the World Wide Web*, 23:2–15, 2013.
17. Apache Jena. Reasoners and rule engines: Jena inference support. *The Apache Software Foundation*, 2013.
18. Timothy Lebo, Satya Sahoo, Deborah McGuinness, et al. Prov-o: The prov ontology. *W3C Recommendation*, 30, 2013.
19. Simon Miles, Paul Groth, Miguel Branco, and Luc Moreau. The requirements of using provenance in e-science experiments. *Journal of Grid Computing*, 5(1):1–25, 2007.
20. Paolo Missier, Khalid Belhajjame, and James 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.
21. Kuhn T, Chichester C, Krauthammer M, et al. Decentralized provenance-aware publishing with nanopublications. *PeerJ Computer Science 2:e78 https://doi.org/10.7717/peerj-cs.78*.
22. Katherine Wolstencroft, Robert Haines, Donal 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.
23. Jun Zhao, Chris Wroe, Carole Goble, et al. Using semantic web technologies for representing e-science provenance. In *The Semantic Web–ISWC 2004*, pages 92–106. Springer, 2004.