# Provenance for Lattice QCD Workflows – An Update


Tanja Auge
University of Regensburg
Regensburg, Germany
tanja.auge@ur.de

Gunnar Bali
Univeristy of Regensburg
Regensburg, Germany
gunnar.bali@ur.de

Christian Kindler
University of Regensburg
Regensburg, Germany
christian.kindler@ur.de

Meike Klettke
University of Regensburg
Regensburg, Germany
meike.klettke@ur.de

Daniel Knüttel
University of Regensburg
Regensburg, Germany
daniel.knuettel@ur.de

Wolfgang Söldner
University of Regensburg
Regensburg, Germany
wolfgang.soeldner@ur.de

Tilo Wettig
University of Regensburg
Regensburg, Germany
tilo.wettig@ur.de



## Abstract

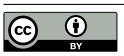
Particle physics research seeks to understand the fundamental constituents of matter and their interactions. As part of the standard model of particle physics, *Quantum Chromodynamics (QCD)* is the quantum field theory underlying the strong interactions. This theory explains the formation of hadrons, i.e., composite particles such as the proton and neutron found in atomic nuclei. *Lattice QCD* is a numerical formulation of QCD, and its development over decades has provided crucial insights into the internal structure of hadrons. The workflows of Lattice QCD involve computationally immense tasks including the generation and manipulation of large amounts of data. A proper description of these workflows and the tracking of the associated metadata is essential and allows for the validation of results at any stage.

We have previously established a provenance model for the two generic parts of the Lattice QCD workflow (*generation* and *measurement*). The model is based on the W3C PROV standard. In this paper, we again follow this standard and extend our model by the third part of the workflow (the *analysis* part). For the generation part of the workflow a community metadata standard (*QCDml*) exists, which includes provenance information. There are no uniform standards for the other two parts. As a first step towards such standards, we introduce the concept of a *ProvCard* to isolate the provenance-related metadata. ProvCards allow us to answer typical provenance questions quickly and independently of other (meta-) data. Furthermore, we can automatically generate provenance graphs from ProvCards. We present an outline of the toolbox `provQCD`, which will implement the corresponding functionalities.



## CCS Concepts

• **Information systems** → **Data provenance**.



*PW' 25, Berlin, Germany*



ACM ISBN 979-8-4007-1941-7/25/06
https://doi.org/10.1145/3736229.3736268

## Keywords

Workflow provenance, W3C PROV, Metadata standards, Lattice QCD

**ACM Reference Format:**
Tanja Auge, Gunnar Bali, Christian Kindler, Meike Klettke, Daniel Knüttel, Wolfgang Söldner, and Tilo Wettig. 2025. Provenance for Lattice QCD Workflows – An Update. In *ProvenanceWeek (PW' 25), June 22–27, 2025, Berlin, Germany*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3736229.3736268


## 1 Introduction

*Lattice QCD* is a well-established numerical formulation of *Quantum Chromodynamics (QCD)* — the theory of quarks and gluons. In this approach, QCD is formulated on a four-dimensional spacetime lattice, resulting in high-dimensional integrals that cannot be solved analytically but are evaluated numerically to extract physical results, which are referred to as observables. Such results can then be compared to real-world experiments, e.g., at high-energy colliders like the LHC at CERN. The overall aim is to answer fundamental questions, e.g., how the properties of a proton such as mass and spin follow from the interactions of its constituents – the quarks and gluons.

Due to the complexity of the theory large computational and storage resources are needed to obtain Lattice QCD results at the required precision. The sizes of typical datasets have grown to about 1 PetaByte of primary data and several PetaBytes of derived data, and they keep growing. Thus, it becomes increasingly important to improve the data and workflow management, including provenance.

A typical Lattice QCD workflow generally factorizes into three parts: *generation*, *measurement*, and *analysis* [1, 3]. In the first part, *ensembles* of gauge-field *configurations* (the primary data) are generated. In the second part, *correlation functions* are computed from these configurations. In the third part, the correlation functions are combined into the observables of interest.

Provenance as a field supports researchers in their computational-science work by addressing questions such as

**Q1** Which datasets are affected by an error or bug?
**Q2** How are datasets affected by modifying a parameter?

**Q3** Who was involved in generating the data?

To answer such questions provenance-related metadata need to be collected throughout the workflow (for every code execution and every dataset).

To integrate provenance into Lattice QCD workflows a sustainable provenance model is needed. A first version of a W3C PROV model for the generation and the measurement part was introduced in [2]. In that model the analysis part was still missing. This part is less generic than the first two parts and varies between (and even within) collaborations.

***Contributions.*** We extend the W3C PROV model introduced in [2] by the analysis part. We introduce the *ProvCard* concept to isolate provenance-related metadata. From the ProvCards, we generate a *ProvGraph* in a PROV-DM format [9] to visualize the workflow provenance and answer typical provenance questions. Finally, we present an implementation outline of a toolbox to generate ProvCards and ProvGraphs.

## 2 Background

Lattice QCD involves complex workflows in which large amounts of data are generated and processed using a variety of (high-performance) computing resources. It is therefore important to establish a suitable provenance model that allows for the automated generation and subsequent analysis of relevant provenance information. Workflow provenance is usually modeled using the W3C PROV standard [7]. In this standard, it is described by graphs whose nodes are *entities*, *activities*, or *agents* (cf. Figure 1 below). Entities are data or artifacts and can be *derived from* other entities. Activities can *generate* or *use* entities. Agents can perform or control activities (*wasAssociatedWith*) or produce entities (*wasAttributedTo*).

Provenance information can be extracted from metadata. Metadata handling usually differs between the three parts of the Lattice QCD workflow and between collaborations. The most commonly used formats are XML, JSON, or just text files. For the first part of the workflow (generation) there is a community-wide metadata standard, *QCDml* [5, 8, 11], which also includes provenance information, see Section 3 below. For the remaining two parts, community metadata standards have not been established yet.

## 3 Community Metadata Standard

Lattice QCD gauge-field configurations are computationally expensive to produce. It is therefore sensible to share ensembles of configurations within the community, adhering to the FAIR principles (Findability, Accessibility, Interoperability, Reusability) [12]. Findability can be provided by a database that enables users to search for (meta-) data. Accessibility is obtained if (meta-) data have unique identifiers and can be retrieved using standard communication protocols. Interoperability is based on common data and metadata formats. These principles are realized in the *International Lattice Data Grid (ILDG)* [6, 8, 10], a federation of several autonomous regional grids. The *Metadata Working Group* is responsible for developing community-wide metadata schemas and the file format. The *Middleware Working Group* defines interfaces for the interoperability of services and infrastructures in different regions.

The ILDG has established a Lattice QCD gauge-field configuration file format, an *ensemble metadata* standard, and a *configuration metadata* standard [5, 11]. Both *QCDml* metadata standards[1] also include provenance information. For example, the ensemble metadata include agents that added, revised, or removed an ensemble, while the configuration metadata include the agent who generated the configuration. A unique identifier, the *Markov Chain URI*, is used to link ensemble and configuration metadata.

In every regional grid that is part of ILDG, the metadata are stored in a database, the *Metadata Catalogue (MDC)*, and can be searched using XPath queries or *quick-search keys*. The latter [11] offer faster but more limited search capabilities. The latest specifications and standards are summarized in ILDG 2.0 [6].

## 4 Provenance for Lattice QCD

In this section, we discuss the provenance-related aspects of the three parts of a generic Lattice QCD workflow introduced above: *generation*, *measurement*, and *analysis*. In [2], we focused on the first two compute-intensive parts. Here we focus on the third part.

### 4.1 Generation and Measurement Parts

In the generation part, gauge-field configurations are generated using Markov-chain Monte-Carlo techniques and stored for further analysis. In the measurement part, correlation functions and other derived data are computed from these configurations. A main difference between these two parts is that generation requires long sequential runs since every configuration depends on the previous one, while in the second part many measurements can be carried out independently in parallel. This is reflected in the corresponding metadata.

In our previous paper [2], we described the workflow for the generation and measurement parts and proposed a suitable provenance model. This is shown in the upper two parts of Figure 1. Simulation parameters, including physical and algorithmic parameters, serve as input for the generation part, while measurement parameters provide input for the measurement part. Metadata are generated along with the data.

### 4.2 Analysis Part

Analysis is the final part of the Lattice QCD workflow. This part is less generic than the other two parts because there are many different observables that can be derived from the same set of correlation functions, i.e., from the output of the measurement part. Furthermore, independent codes are frequently developed for the same observable by one and the same research group for verification purposes. Nevertheless, the analysis part includes certain elements that are common to the workflows of most, if not all, collaborations. To capture these elements, we divide the analysis part into three *stages*: *extraction*, *ensemble averaging*, and *combination*.

Typically only a subset of correlation functions is of interest for a particular analysis. In the *extraction stage*, this subset is extracted from the measurement data for subsequent use in the following stage. The *ensemble-averaging stage* consists of averaging, statistical analysis, and fitting for one ensemble of the extracted correlation functions. In the *combination stage*, the ensemble averages are combined to determine the physical observables of interest. This stage again involves statistical analysis and fitting.

[1] https://gitlab.desy.de/ildg/mdwg/qcdml/

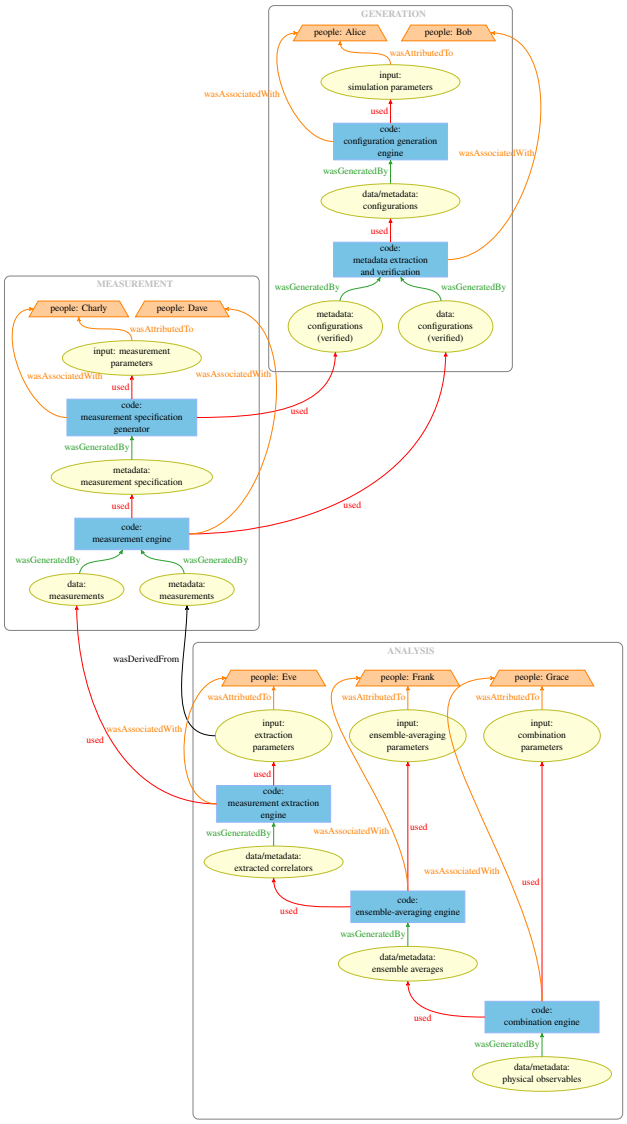



**Figure 1: Simplified version of a W3C PROV model for a Lattice QCD workflow consisting of generation, measurement, and analysis parts. Yellow, blue, and orange nodes correspond to entities, activities, and agents, respectively.**

The task now is to derive a provenance model for the analysis part. Because of the non-generic nature of this part, it is unrealistic to pursue the goal of a universal model that would be applicable to the workflows of all research groups. Instead, we focus on the analysis part of our specific workflow and extend our previous provenance model [2] to include this part. We believe that both the resulting model and our prototype implementation (see Section 5 below) are sufficiently generic to serve as a blueprint for other research groups.

In the W3C PROV standard, a PROV data model (PROV-DM, [9]) is represented by a provenance graph, i.e., there is a one-to-one correspondence between model and graph. To not overload the figure, we have simplified our extended provenance model. This simplified version is shown in Figure 1. It differs from the full model in two aspects: (1) We have sometimes combined data and metadata into one entity. (2) We have assumed that the agent whom an input file is attributed to is also associated with the execution of the activity which is using that input file.

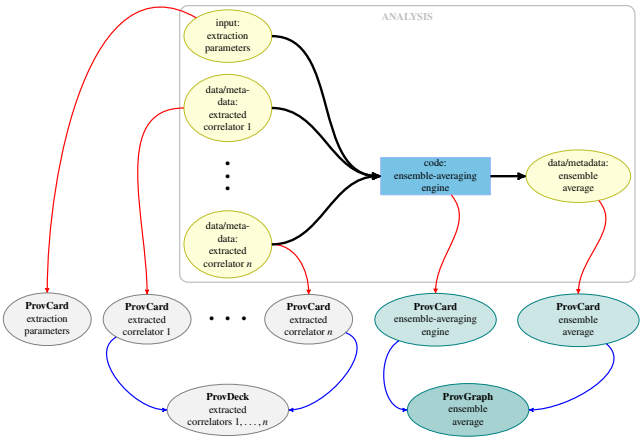



**Figure 2: Provenance concept for a subset of the analysis part. The gray and green nodes extend the workflow by provenance information. The red arrows indicate the generation or extraction of ProvCards from the source's metadata, while the blue arrows indicate the generation of ProvDecks and ProvGraphs from ProvCards. Both functionalities are part of the provQCD toolbox, see Section 5.**

## 4.3 ProvCards, ProvDecks, and ProvGraphs

To respect the FAIR principles F1–F3 [12] we represent every entity, activity, and agent as a triple consisting of (1) a unique and persistent identifier, (2) the metadata, and (3) the actual content (e.g., a dataset or a code). To be able to answer provenance questions quickly and independently, we wish to isolate and store the provenance-related subset of the metadata in such a way that a workflow provenance graph (*ProvGraph*) can be generated automatically in a PROV-DM format [9].

For a given entity, activity, or agent, we call the provenance-related metadata a *ProvCard* and store this ProvCard in a file or database. Ideally, the ProvCard is generated as the workflow is executed, but if this was not the case in past workflow executions it will be necessary to extract the ProvCard from existing metadata or log files.

In the following we refer to the entity, activity, or agent from which the ProvCard was generated or extracted as the *source*. Every ProvCard contains the unique persistent identifier of the source, a unique and persistent identifier of the ProvCard (which could be derived from the identifier of the source), the type (of the source), and a comment. The remaining entries of a ProvCard depend on the type of the source and include, e.g., the associated/attributed agent or input files (code and data files).

The ProvCards are registered in a searchable database (FAIR principle F4). In most situations it will be convenient to organize the ProvCards in some way, for which we introduce the concept of a *ProvDeck* as a collection of ProvCards. Depending on the use case, it may or may not be sensible to define a hierarchy of ProvDecks.

In Figure 2 we illustrate our provenance concept for a subset of the analysis part. As an example for a typical use case, the entity "extracted correlators" from Figure 1 is split into $n = 291$ entities corresponding to the correlators from the 291 different configurations we analyze. The corresponding 291 ProvCards are collected in a ProvDeck. To generate the ProvGraph for Figure 2 we only need the two ProvCards for the ensemble-averaging engine and the ensemble average.

Figure 2 assumes a toolbox `provQCD` whose implementation is outlined in Section 5. This toolbox will have two main functionalities: it extracts the ProvCard from the corresponding source (if the ProvCard was not generated by the workflow), and it generates the ProvGraphs from the ProvCards. Of course, both the provenance concept and the toolbox can be applied to all parts of the workflow.

For the ProvGraph any PROV-DM format can be used. For definiteness and better readibility we will use PROV-JSON [4] for the ProvGraph and JSON for the ProvCards. A ProvCard then corresponds to a single JSON object. In Listings 1 and 2 we show the two ProvCards that are needed to generate the ProvGraph for Figure 2. The resulting ProvGraph is shown in Listing 3 and Figure 3.

**Listing 1: ProvCard for an Ensemble Average Entity**

```
{
  "id": "prov::data://ENSEMBLE$_ID/PROJ/nuc_mass.npy",
  "source_id": "data://ENSEMBLE_ID/PROJ/nuc_mass.npy",
  "comment": "nucleon mass on ...",
  "type": "provQCD:entity:data",
  "input": [
    {
      "id": "prov::activity://ENSEMBLE_ID/PROJ/nm_extract.
            py@GIT_HASH/1743601429",
      "source_type": "provQCD:activity",
      "edge_type": "wasGeneratedBy"
    }
  ]
}
```

**Listing 2: ProvCard for an Ensemble Averaging Activity**

```
{
  "id": "prov::activity://ENSEMBLE_ID/PROJ/nm_extr.py@GIT_HASH
        /1743601429",
  "source_id": "activity://ENSEMBLE_ID/PROJ/nm_extr.py@GIT_HASH
        /1743601429",
  "comment": "nucleon mass on ...",
  "type": "provQCD:activity",
  "attributedTo": "Franks_ORCiD",
  "code_id": "prov::code://ENSEMBLE_ID/PROJ/nm_extr.py@GIT_HASH",
  "input": [
    {
      "id": "prov::data://ENSEMBLE_ID/PROJ/nm_params.dat",
      "source_type": "provQCD:entity:input",
      "edge_type": "used"
    },
    {
      "id": "prov::data://ENSEMBLE_ID/PROJ/corr.1.h5",
      "source_type": "provQCD:entity:data",
      "edge_type": "used"
    },
    ...
    {
      "id": "prov::data://ENSEMBLE_ID/PROJ/corr.291.h5",
      "source_type": "provQCD:entity:data",
      "edge_type": "used"
    }
  ]
}
```

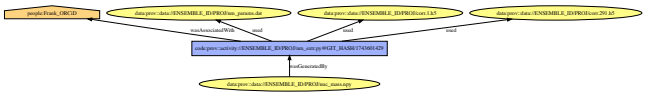


**Figure 3: ProvGraph for an Ensemble Average (automatically rendered from Listing 3 using ProvViz[2]).**

The availability of ProvCards and ProvGraphs allows us to answer a large number of typical provenance questions, such as **Q1** to **Q3**. Note that the ProvCards contain the full provenance information that was generated or extracted, while the ProvGraphs can provide a very useful visualization.

## 5 Implementation Outline

In the preceding section we have introduced ProvCards that contain provenance-relevant metadata. If the ProvCards were not generated automatically during workflow execution, the required information needs to be extracted from metadata or log files. Furthermore, we want to generate ProvGraphs from ProvCards. For these purposes we are developing the toolbox `provQCD`, which consists of several modules whose functionalities are outlined below.

- If ProvCards were not generated during workflow execution, the `metadata_extractor` module extracts the provenance-related metadata from metadata or log files.
- As the entire Lattice QCD workflow is fragmented and carried out successively by various agents who are not necessarily familiar with the W3C PROV standard, the `translator` module translates the output of the `metadata_extractor` module into W3C PROV-compliant terminology and generates a proper ProvCard.
- If desired, the `merger` module collects ProvCards in a ProvDeck.
- The `registrar` module registers ProvCards in a searchable database.
- The `graph_generator` module generates a ProvGraph from ProvCards or ProvDecks, in a PROV-DM format such as PROV-JSON or PROV-XML.
- The `visualizer` module creates a visual representation of the ProvGraph.

The `provQCD` toolbox will incorporate functionalities of the existing W3C PROV-compliant Python package `prov`.[3] Also, it will adhere to ILDG standards and FAIR principles.

## 6 Conclusion and Future Work

We have extended the W3C PROV model introduced in [2] to the third part of the Lattice QCD workflow, analysis. This part is less universal than the other two parts but does include generic elements. Therefore we hope that other research groups can adapt our provenance model in a straightforward manner.

We have introduced the concept of ProvCards to capture the provenance-related aspects of the metadata of activities, entities, and agents. The advantage of ProvCards is that they allow us to answer provenance questions without having to search through large amounts of (meta-) data. From the ProvCards we can generate

[2] https://provviz.com/
[3] https://pypi.org/project/prov

a ProvGraph for the workflow, or parts thereof, which can provide a useful visual aid. We are developing the `provQCD` toolbox, whose main purposes are the extraction of ProvCards from metadata and log files as well as the generation of ProvGraphs from ProvCards.

Future work includes the specification of the metadata standard for the ProvCards and the corresponding ontology for the Lattice QCD community as well as the completion of the `provQCD` toolbox.

## Acknowledgments

We thank Marcel Rodekamp and Hubert Simma for stimulating discussions. This work was supported in part by the Deutsche Forschungsgemeinschaft (DFG) under grant PUNCH4NFDI, project number 460248186.

## References

[1] Yasumichi Aoki et al. 2025. Lattice gauge ensembles and data management. *PoS* LATTICE2024 (2025), 412. doi:10.22323/1.466.0412 arXiv:2502.08303 [hep-lat]

[2] Tanja Auge, Gunnar Bali, Meike Klettke, Bertram Ludäscher, Wolfgang Söldner, Simon Weishäupl, and Tilo Wettig. 2023. Provenance for Lattice QCD workflows. In *WWW (Companion Volume)*. ACM, 1524–1530.

[3] Gunnar Bali et al. 2022. Lattice gauge ensembles and data management. *Proceedings of Science (PoS)* LATTICE (2022), 203. arXiv:2212.10138

[4] Trung Dong Huynh, Michael O. Jewell, Amir Sezavar Keshavarz, Danius T. Michaelides, Huanjia Yang, and Luc Moreau. 2013. The PROV-JSON Serialization. https://www.w3.org/submissions/prov-json/.

[5] International Lattice Data Grid (ILDG). 2025. Specification of ILDG Standards. https://hpc.desy.de/ildg/specifications/.

[6] Frithjof Karsch, Hubert Simma, and Tomoteru Yoshie. 2022. The International Lattice Data Grid - towards FAIR Data. In *LATTICE 2022*. arXiv:2212.08392 [hep-lat]

[7] Timothy Lebo et al. 2013. PROV-O: The PROV Ontology. https://www.w3.org/TR/prov-o/.

[8] C. M. Maynard and D. Pleiter. 2005. QCDml: First milestone for building an International Lattice Data Grid. *Nucl. Phys. B Proc. Suppl.* 140 (2005), 213–221. doi:10.1016/j.nuclphysbps.2004.11.116 arXiv:hep-lat/0409055

[9] Luc Moreau et al. 2013. PROV-DM: The PROV Data Model. https://www.w3.org/TR/2013/REC-prov-dm-20130430/.

[10] Giovanni Pederiva, Basavaraja Bheemalingappa Sagar, Daniel Knüttel, and Christian Schmidt. 2025. Enhanced Data Management Tools for ILDG and PUNCH4NFDI. In *LATTICE 2024*, Vol. 466. 450. doi:10.22323/1.466.0450

[11] Basavaraja Bheemalingappa Sagar, Georg von Hippel, Giannis Koutsou, Hideo Matsufuru, Dirk Pleiter, Hubert Simma, and Carsten Urbach. 2025. International Lattice Data Grid 2.0: Status and Progress. *PoS* LATTICE2024 (2025), 411. doi:10.22323/1.466.0411 arXiv:2502.09253 [hep-lat]

[12] Mark D. Wilkinson et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3, 1 (2016), 160018. doi:10.1038/sdata.2016.18

# Appendix

**Listing 3: ProvGraph following from Listings 1 and 2**

```
{
  "prefix": {
    "people": "http://example.org/provQCD",
    "data": "http://example.org/provQCD",
    "code": "http://example.org/provQCD"
  },
  "agent": {
    "people:Frank": {}
  },
  "entity": {
    "data:prov::data://ENSEMBLE_ID/PROJ/nm_params.dat": {},
    "data:prov::data://ENSEMBLE_ID/PROJ/nuc_mass.npy": {},
    "data:prov::data://ENSEMBLE_ID/PROJ/corr.1.h5": {},
    "data:prov::data://ENSEMBLE_ID/PROJ/corr.291.h5": {}
  },
  "activity": {
    "code:prov::activity://ENSEMBLE_ID/PROJ/nm_extr.py@GIT_HASH
        /1743601429": {}
  },
  "wasGeneratedBy": {
    "_:id1": {
      "prov:entity": "data:prov::data://ENSEMBLE_ID/PROJ/nuc_mass.
          npy",
      "prov:activity": "code:prov::activity://ENSEMBLE_ID/PROJ/
          nm_extr.py@GIT_HASH/1743601429"
        }
  },
  "used": {
    "_:id2": {
      "prov:activity": "code:prov::activity://ENSEMBLE_ID/PROJ/
          nm_extr.py@GIT_HASH/1743601429",
      "prov:entity": "data:prov::data://ENSEMBLE_ID/PROJ/corr.1.h5
          "
    },
    "_:id3": {
      "prov:activity": "code:prov::activity://ENSEMBLE_ID/PROJ/
          nm_extr.py@GIT_HASH/1743601429",
      "prov:entity": "data:prov::data://ENSEMBLE_ID/PROJ/corr.291.
          h5"
        },
    "_:id4": {
      "prov:activity": "code:prov::activity://ENSEMBLE_ID/PROJ/
          nm_extr.py@GIT_HASH/1743601429",
      "prov:entity": "data:prov::data://ENSEMBLE_ID/PROJ/nm_params
          .dat"
    }
  },
  "wasAssociatedWith": {
    "_:id5": {
      "prov:activity": "code:prov::activity://ENSEMBLE_ID/PROJ/
          nm_extr.py@GIT_HASH/1743601429",
      "prov:agent": "people:Frank_ORCiD"
    }
  }
}
```