Mobilizing Bat Literature

using versioned snapshots of the Zotero BatLit Library

Jorrit Poelen (UC Santa Barbara Cheadle Center, Ronin Institute, GloBI)

2025-08-26

Guiding Questions

How to manage the BatLit corpus?

How to create versioned snapshots of the BatLit corpus?

How to share versioned snapshots of the BatLit corpus?

Guiding Questions - Brief Answers

How to manage the BatLit corpus?

We use Zotero to manage our literature corpus.

How to create versioned snapshots of the BatLit corpus?

We use Preston to version our literature corpus [1].

How to share versioned snapshots of the BatLit corpus?

We use Zenodo to allow versioned access to BatLit.

---

[1] Elliott M.J., Poelen, J.H. & Fortes, J.A.B. (2023) Signing data citations enables data verification and citation persistence. *Sci Data*. https://doi.org/10.1038/s41597-023-02230-y hash://sha256/f849c870565f608899f183ca261365dce9c9f1c5441b1c779e0db49df9c2a19d

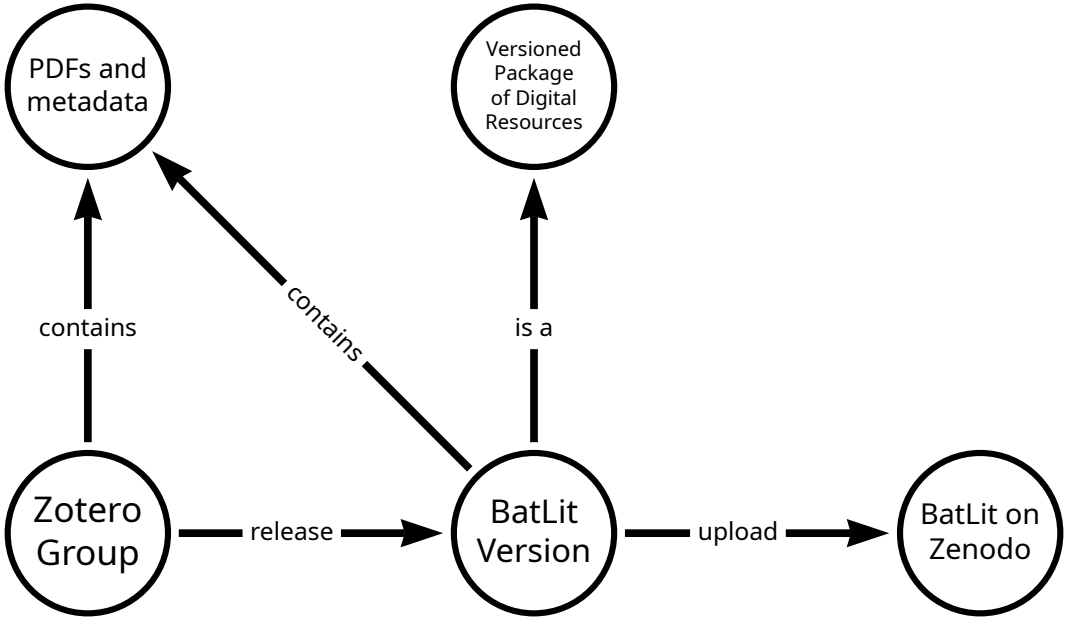
Guiding Questions - More Complete Answers

The BatLit Data Paper describes our BatLit workflow and provides specific examples.

Also, the paper relies on Zotero and Zenodo documentation to answer any questions about these platforms.

The following sections help you get started on Preston and their relation to Zotero and Zenodo.

BatLit Publication Workflow



Why Preston?

Preston allows for creating a versioned snapshot of a Zotero library (or group).

Also, Preston allows for depositing this versioned snapshot into Zenodo.

At time of writing, 26 Aug 2025, this publish Zotero-to-Zenodo functionality only offered through Preston.

Part I - Preston Basics

1. Tracks (or versions) *Digital* Content
2. Format Agnostic (*any* Digital Content)
3. Allows Signed Data Citations

Prequisites

1. Familiarity with Unix Shell
2. Access to a computer with some Unix Shell
3. Install Preston [2]

[2] https://globalbioticinteractions.org/preston

Check Preston Version

run

```
preston --version
```

and verify that this produces something like:

```
0.11.0
```

Say Hi and Version It

Copy and paste this into your commandline.

```
mkdir -p some/empty-directory
cd some/empty-directory
echo hi there! | preston track --algo md5 | grep hasVersion | preston cat
```

which should produce...

```
hey there!
```

Step I.1 Say Hi and Version It

```
echo hi there!
```

Prints “hi there!” to standard output

Step I.2 Say Hi and Version It

```
echo hi there! | preston track --algo md5
```

Print "hi there!" to output (stdout), then turn this output into input (stdin) of `preston track`. So, this sends "hi there!" to preston and versions (or tracks) the input. The output of preston is a machine readable description of what happened and ends with something like:

```
<...> <...hasVersion> <hash://md5/75c7e31591354f2c82226aa3eb0267c7> <...>
```

This preston output, or description, is formatted in rdf/nquads and records what content was recorded when and by who. This machine readable description is also known as the BOM Bill of Materials, manifest or packing slip for the tracked content.

Step I.3 - Say Hi and Version It

```
echo hi there! | preston track --algo md5 | grep hasVersion
```

This prints only the part of the BOM that includes “hasVersion” and should look something like:

```
<urn:uuid:X> <...hasVersion> <hash://md5/75c7e31591354f2c82226aa3eb0267c7>
```

Step I.4 - Say Hi and Version It

```
<urn:uuid:X> <...hasVersion> <hash://md5/75c7e31591354f2c82226aa3eb0267c7>
```

This is a statement expressed in rdf/nquad. In this case, it expressed something like: there's this thing `urn:uuid:X` that is associated with content that has a cryptographic hash `hash://md5/75c7e31591354f2c82226aa3eb0267c7`. A cryptographic hash is a unique fingerprint derived from the digital content itself. **If the content and the hash algorithm are the same, the fingerprint is always the same.** This concept is central to internet security as well as things like cryptocurrencies.

Step I.5 - Say Hi and Version It

Now, we ask Preston to print the versioned content by piping the "hasVersion" statement into "preston cat"':

```
echo hi there! | preston track --algo md5 | grep hasVersion | preston cat
```

to produce . . .

```
hi there!
```

Step I.6 - Say Hi and Version It

Now that we've versioned saying hi, we can print the content using

```
preston cat hash://md5/75c7e31591354f2c82226aa3eb0267c7
```

to produce . . .

```
hi there!
```

This suggests that Preston *thinks* that `hash://md5/75c7e31591354f2c82226aa3eb0267c7` is the cryptographic hash of `hi there!`. And . . .

Step I.7 - Say Hi and Version It

If you know the fingerprint of content (e.g., `hash://md5/75c7e31591354f2c82226aa3eb0267c7`), you can use it to ask for what *exactly* what you want. And, on getting a result, you can *independently* verify that this is the case using some commonly available cryptographic hash calculators like `md5sum` (linux) or `md5` (Mac). These calculators are readily available as they are central to internet security and other core applications.

Step I.7 - Say Hi and Version It Continued...

```
preston cat hash://md5/75c7e31591354f2c82226aa3eb0267c7 | md5sum
```

produces:

```
75c7e31591354f2c82226aa3eb0267c7  -
```

Showing that an independent tool `md5sum` verified that the content you asked for is the content you got!

Takeaways

- Preston tracks, versions and packages digital content
- Cryptographic hashes are unique digital fingerprints for digital content.
- Cryptographic hashes can be generated independently using commonly available tools.
- Cryptographic hashes enable secure citation of digital content

Next Steps

- Review BatLit Datapaper
- Create a Zotero Group for Testing
- Track the Zotero Test Group using Preston
- Create a Zenodo Test Community on Zenodo Sandbox
- Publish the Zotero Test Group to the Zenodo Test Community
- Once you feel comfortable, repeat the process with the "real" Zotero BatLit Group and associated Zenodo BatLit Community

Part II - Creating a Snapshot of a Zotero Literature Group

After covering the Preston basics, we now explore how to create a versioned snapshot of a Zotero Literature Collection.

First, we create a Zotero Group for Testing, then we track the Zotero Test Group using Preston

Step II.1- Create A Zotero Group for Testing

1. go to https://zotero.org
2. login using your credentials
3. create a new empty private Zotero group for testing
4. download pdfs associated with https://doi.org/10.5281/zenodo.13418040 and https://doi.org/10.5281/zenodo.14817268
5. import the two pdfs into your Zotero Test Group

Step II.2 - Get your Zotero API Key

In order to talk to Zotero using Preston (or any programmatic method) you need a Zotero Web API Key.

1. search Zotero documentation on getting an API Key

   At time of writing, 2025-08-28, the following web pages

   https://www.zotero.org/settings/keys

   https://www.zotero.org/settings/security#applications

2. Get the API key and record it in a safe location. An API key is a combation of letters and number e.g., 12345678

Step II.3 - Take a snapshot of your Zotero Collection Using Preston

Preston has built in functionality to take a snapshot of a Zotero Collection.

1. open a command-line terminal
2. go to your home directory (e.g., `cd ~`)
3. create new directory (e.g., `mkdir batlit-test`)
4. go into the new directory (e.g., `cd batlit-test`)
5. run the following command

```
export ZOTERO_TOKEN=[SECRET]
preston track https://www.zotero.org/groups/6123963/test_aug
```

list all the content of the metadata from the Zotero group across all Bill of Materials

```
preston ls\
 |  grep hasVersion\
 | grep "https://api.zotero.org/groups/6123963/items/"\
 | grep -v "file/view"\
 | sort\
 | preston cat
```

Step II.4 Make a change and create a new snapshot

1. Change the title of one of the publication in your test Zotero Group
2. Make a new snapshot version by re-running:

```
preston track https://www.zotero.org/groups/6123963/test_aug
```

Step II.5 Compare changes in metadata across snapshot versions

After making a change in a Zotero records, and creating a new snapshot, we can compare the different versions of Bill of Materials associated with these snapshots.

In order to do so, we need to (II.5.1) make a sorted list of all metadata for the most recent Bill of Materials and (II.5.2) make a sorted list of all metadata of a previous Bill of Materials. Finally, (II.5.3) we compare the differences between these metadata snapshots.

Step II.5.1 A sorted list of metadata for most recent Bill of Materials

Create a sorted list metadata statement from the Zotero group for the *most recent* Bill of Materials, and list their content

```
preston head\
 | preston cat\
 | grep hasVersion\
 | grep "https://api.zotero.org/groups/6123963/items/"\
 | grep -v "file/view"\
 | sort\
 | preston cat\
 > most-recent-metadata.txt
```

where 6123963 is the group id number of your Zotero Test Group.

Step II.5.2 A sorted list of metadata for a previous version of the Bill of Materials

List all the content of the metadata from the Zotero group for the *oldest* Bill of Materials and print it to a file

```
preston history\
 | tail -1\
 | preston cat\
 |  grep hasVersion\
 | grep "https://api.zotero.org/groups/6123963/items/"\
 | grep -v "file/view"\
 | sort\
 | preston cat\
 > oldest-metadata.txt
```

where 6123963 is the group id number of your Zotero Test Group.

Step II.5.3 Compare changes across metadata associated with two versions of Bill of Materials

Now that we have the Zotero metadata for the most recent Bill of Materials, as well as a previous version, we can use diff to compare the differences.

```
diff most-recent-metadata.txt oldest-metadata.txt
```

to produce

```
[...]
119c119
<             "title": "Pteropus test",
---
>             "title": "Seasonal roosts and foraging movements of the blac
158c158
<             "dateModified": "2025-08-28T15:41:26Z"
---
>             "dateModified": "2025-08-28T15:04:13Z"
```

In this example the title was changed to `Pteropus test` and the diff reflects this.

Part II - Next Steps

- Create a Zenodo Test Community on Zenodo Sandbox
- Publish the Zotero Test Group to the Zenodo Test Community

Part III - Deposit a Zotero Snapshot into Zenodo

Now that we can make a snapshot of a Zotera group, we'd like to take this snapshot and deposit it into Zenodo. In order to do so, we need to (1) create a test Zenodo community in their sandbox (2) create a Zenodo API key, and (3) upload a snapshot version into Zenodo sandbox using Preston.

Part III.1 - Create a test Zenodo community

Zenodo provides a sandbox to try their platform and experiment.

1. go to https://sandbox.zenodo.org
2. login
3. create a community (e.g., BatLit-Test-20250829)

Part III.2 - Generate A Zenodo API Token

To deposit content into Zenodo programmatically (e.g., using Preston), you need a Zenodo Web API Key / Token.

1. go to https://sandbox.zenodo.org
2. select account > developer (?)
3. generate a new key and record this somewhere safe

Part III.3 - Deposit a Zotero Snapshot in the Zenodo Test Community

1. open a command-line terminal
2. set the evironment variable for Zenodo API access

```
export ZENODO_ENDPOINT=https://sandbox.zenodo.org
export ZENODO_TOKEN=[your secret token]
```

3. generate Zenodo metadata for a specific snapshot version

```
preston head\
 | preston cat\
 | preston zotero-stream\
> zenodo.json
```

4. deposit Zenodo records using the Zenodo metadata

mkdir deposit-logs cat zenodo.json
| preston track –data-dir deposit-logs/data
| preston zenodo –data-dir deposit-logs/data –remote file://$PWD/data/
> zenodo-deposit.log “‘