Graeme Watt (IPPP Durham)

Technical Meeting on Nuclear Data Retrieval, Dissemination, and Data Portals

IAEA Headquarters, Vienna, Austria, 12th November 2024

https://hepdata.net

Email: info@hepdata.net

Forum: hepdata-forum.cern.ch

Follow @HEPData

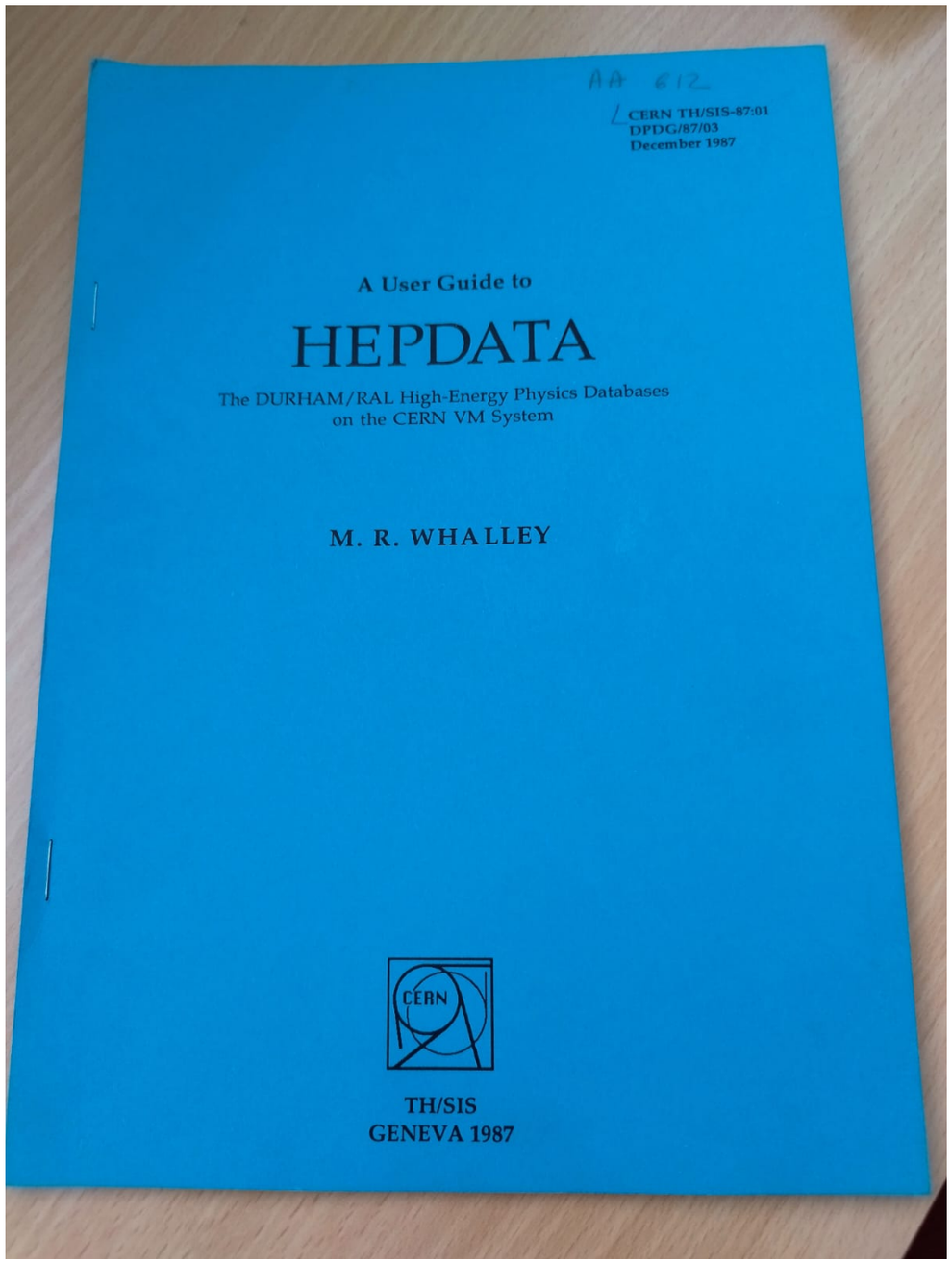
Code: https://github.com/HEPData

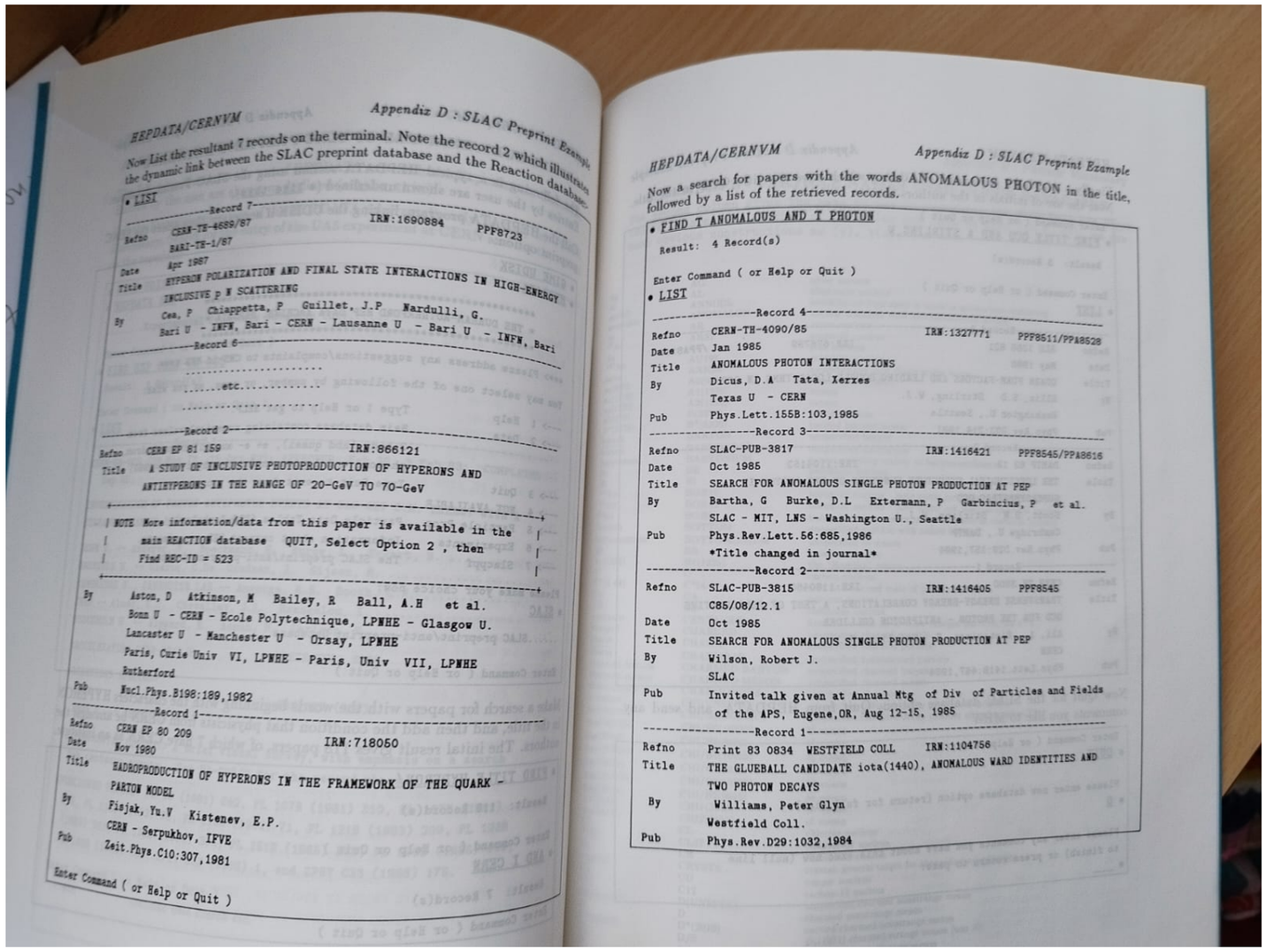
What is HEPData?

- *Open-access* repository for tabular high-level **data** (≲MB) from more than 10k **HEP** publications (138k data tables).
- Interactive *visualisation* and *conversion* to other formats.
- **FAIR** data: **F**indable, **A**ccessible, **I**nteroperable, **R**eusable.
- Funded by UK **S**cience & **T**echnology **F**acilities **C**ouncil.
- Based in **I**nstitute for **P**article **P**hysics **P**henomenology (**IPPP**) at Durham University (UK), going back to 1970s.
- *Current staff in Durham*:
  Graeme Watt (management and user support, since 2013)
  Jordan Byers (software development, since 2022)

HEPData in the 1980s

Berkeley Database Management System (BDMS)





https://cds.cern.ch/record/184048

- Web interface introduced in early 1990s using BDMS + Fortran + CGI scripts.

HEPData redevelopments

hepdata.cedar.ac.uk (switched off in 2022)

- Upgrade 2005-2009 by Andy Buckley and Mike Whalley: "*HepData reloaded: reinventing the HEP data archive*", PoS ACAT2010 (2010) 067 [arXiv:1006.0517].
- Relational database (MySQL) and Java web interface.

hepdata.net (new production site since 2017)

- Partnership with CERN Scientific Information Service.
- Complete rewrite in 2015-2016 by Eamonn Maguire.
- J. Phys.: Conf. Ser. 898 102006 [arXiv:1704.05473]

HEPData infrastructure

- All provided by CERN IT with support from CERN SIS.
- Migration in 2020 from Puppet VMs to Docker/Kubernetes.
- Kubernetes configuration specified in private GitHub repo.
- Argo CD for monitoring and Sentry for error tracking.
- Shared CephFS storage for 1.3M data files (120 GB).
- Database On Demand (DBOD): PostgreSQL v14.10 (6.6 GB).
- OpenSearch v2.15.0 cluster indexes metadata for searching.
- Separate **QA** environment for testing before *production*.
- Discourse instance for Forum: hepdata-forum.cern.ch

HEPData software

- Invenio v3 digital library framework (used by Zenodo).
  - Uses Python 3 and Flask micro web framework.
  - More recent InvenioRDM for turn-key repositories.
- Custom visualisation (using D3.js) and submission code.
- New input text format for data & metadata using YAML.
- Converter from legacy format to YAML (and exporter).
- New data model defined in a PostgreSQL database.
- Metadata indexed with OpenSearch for fast searching.

Code: `https://github.com/HEPData`

https://github.com/HEPData

- `hepdata`: main web application (Python, JavaScript, HTML)
- `hepdata-validator`: JSON schema and validation code
- `hepdata-submission`: documentation and examples
- `hepdata-converter`: YAML to CSV/ROOT/YODA
- `hepdata_lib`: helps transform text/ROOT files to YAML
- `hepdata-cli`: search/download/upload from CLI or API
- `miscellaneous`: Jupyter notebooks for various insights

GitHub Actions workflows used to run automated tests, release Python packages on PyPI and push Docker images to Docker Hub. Dependabot for automatic updates of dependent Python packages.

Modes of data entry

1. Manually harvested from data given in publications.
   HEPData staff extracted tables from `.tex` source.

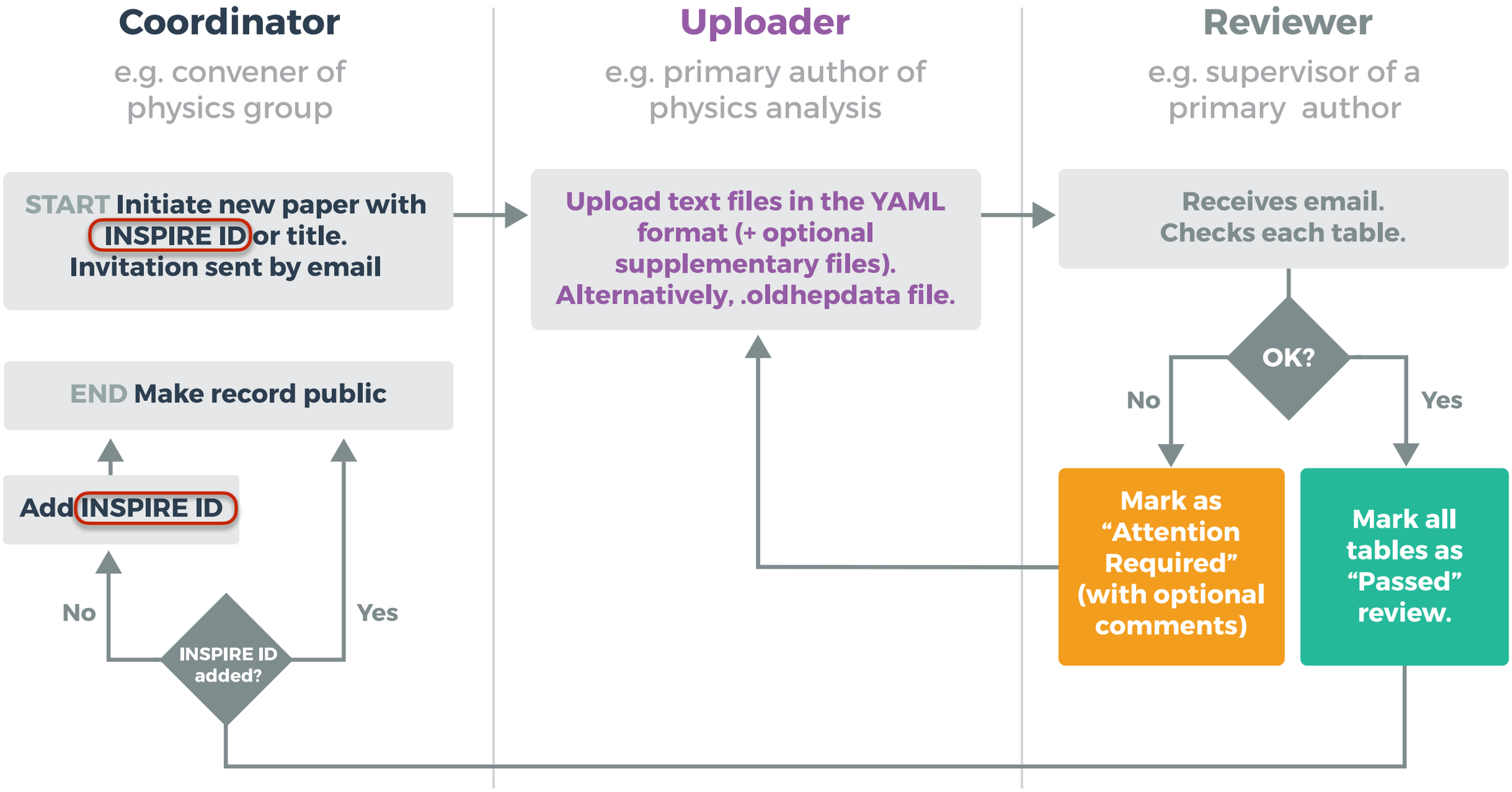
2. Data points directly submitted by experiments.

   - Pre-2014: no guidelines on preferred format.
   - Early 2014: encourage standard "input" format.
   - Late 2014: introduce online submission system.
   - Early 2017: allow submissions from hepdata.net.

Mode 1. now phased out in favour of mode 2.

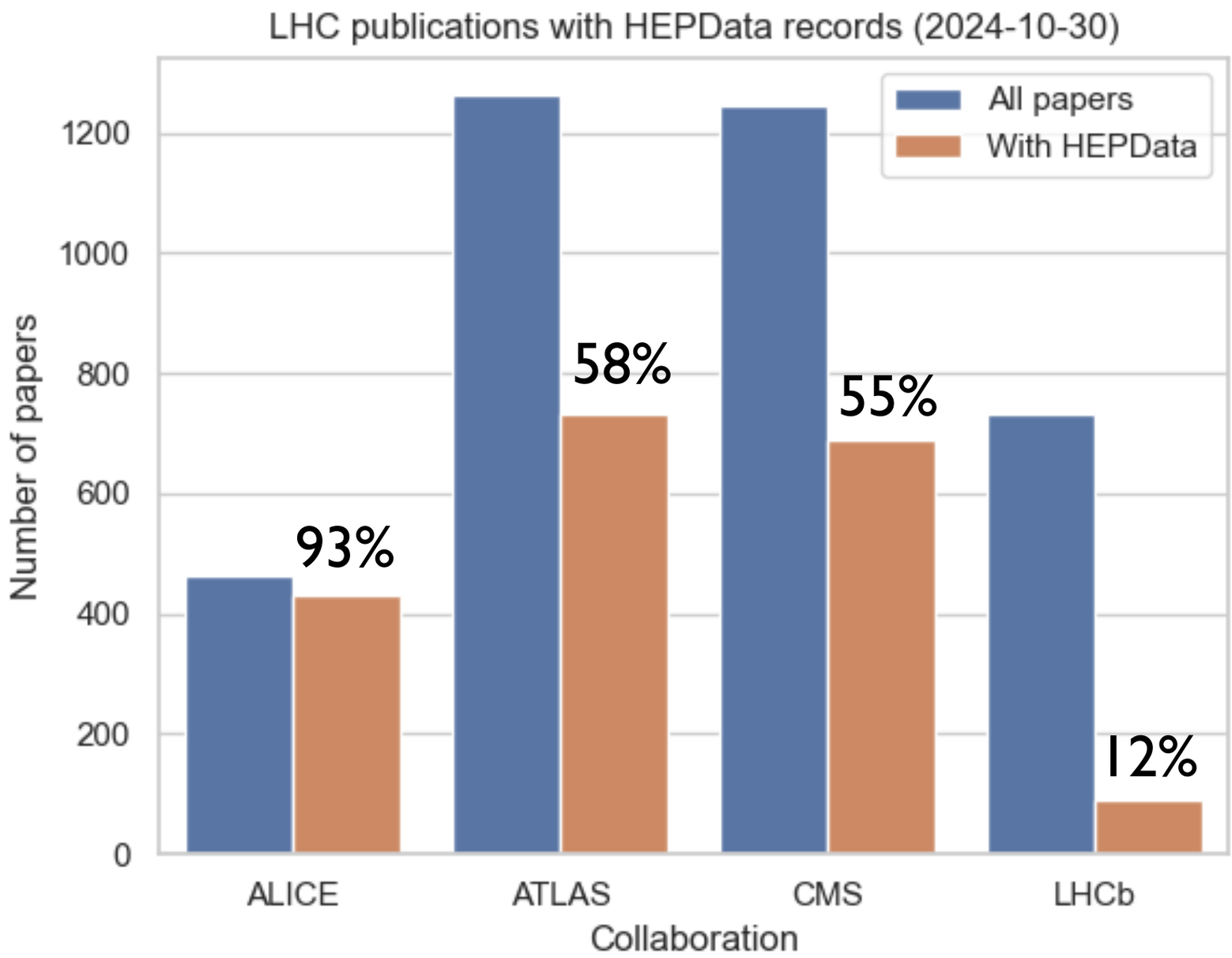
Submission system on hepdata.net

https://hepdata.net/submission



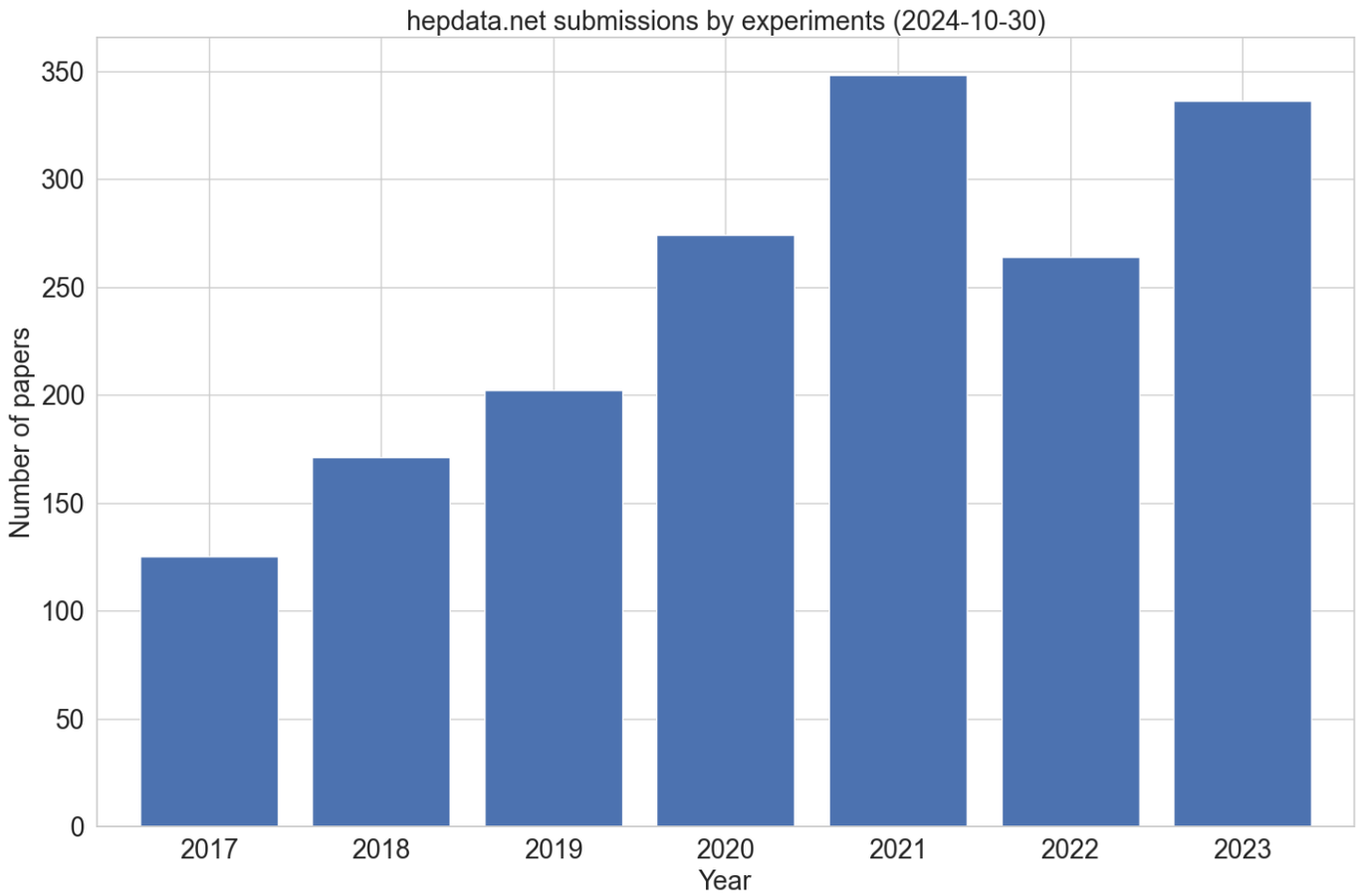
- Submissions managed by Coordinators within each experiment/group.

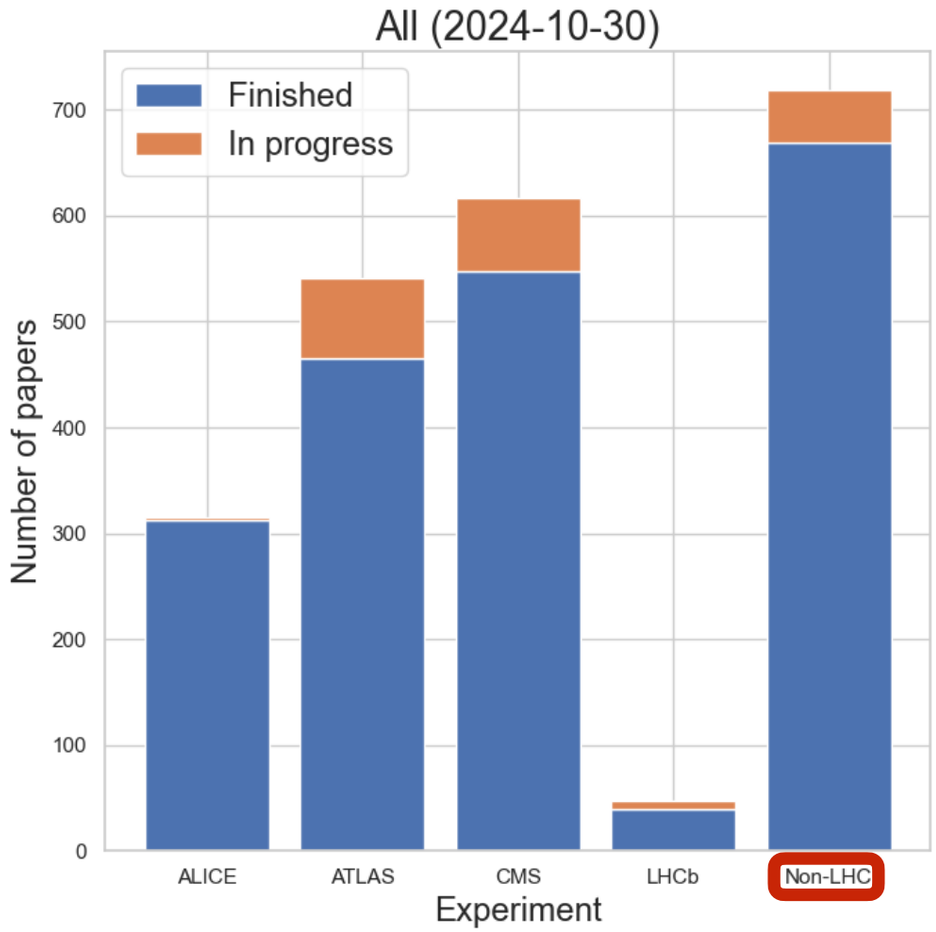
Coverage of LHC publications



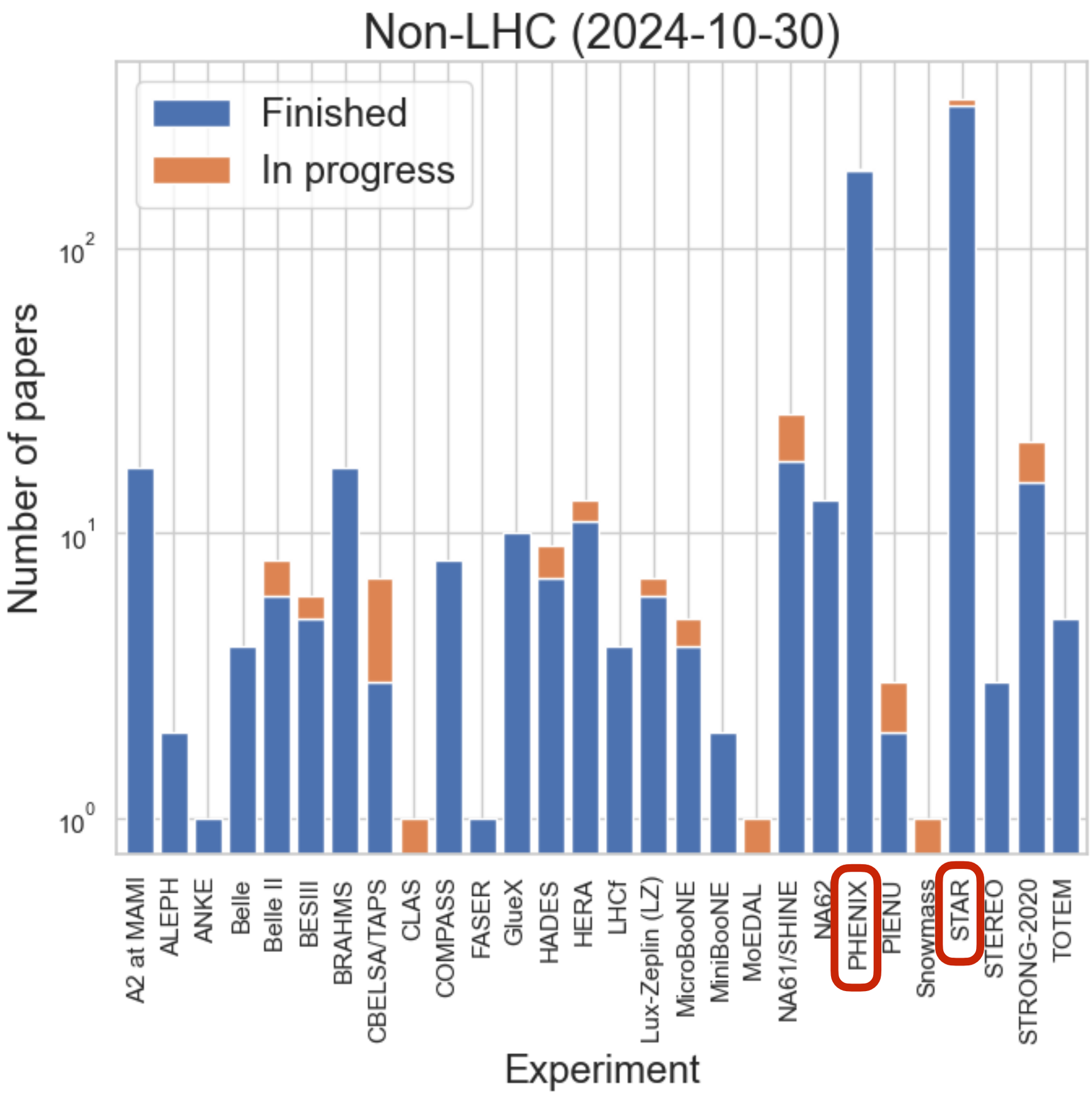
- Search INSPIRE for publications with HEPData (GitHub/Binder).

Submissions via hepdata.net (2017-)





- Increase until 2020, then around 300 submissions/year.
- Large number (669) of completed non-LHC submissions.
- More plots and statistics available in a Jupyter notebook.



- Big efforts by STAR and PHENIX at RHIC (BNL News).

https://www.bnl.gov/newsroom/news.php?a=221570

Physicists and Students Format PHENIX Data for Easy Access

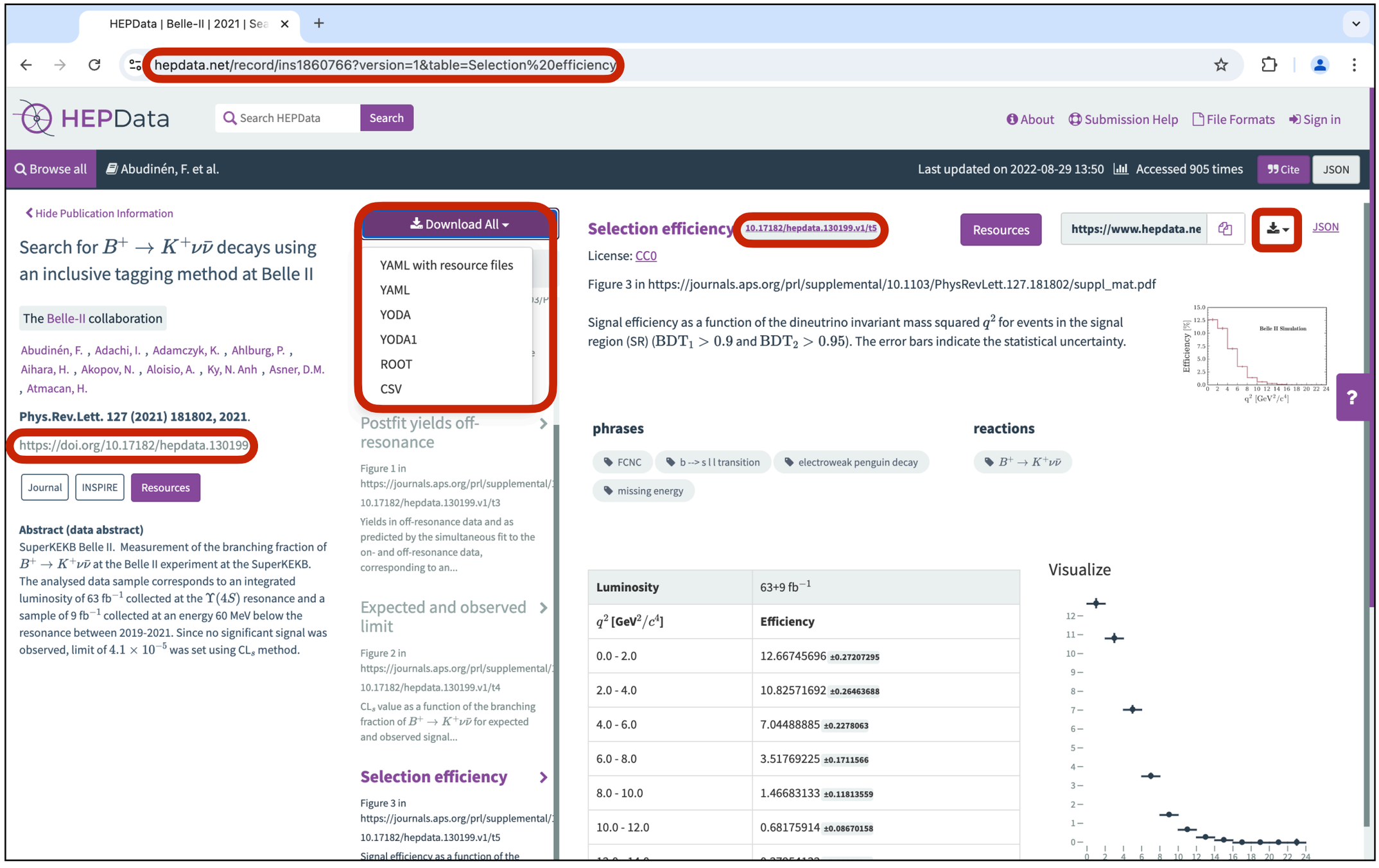
Effort standardizes data needed to unlock the secrets of matter while building skills and bringing new faces into science

November 29, 2023

Christine Nattrass, a physics professor at the University of Tennessee (UT), Knoxville, has recruited a crew of mostly undergraduate students to dig deep into data from billions of particle collisions at the Relativistic Heavy Ion Collider (RHIC)—a U.S. Department of Energy (DOE) Office of Science user facility for nuclear physics research at DOE's Brookhaven National Laboratory. Their goal: reformat data from scientific papers published by RHIC's PHENIX detector collaboration and upload it to a modern database now used across the nuclear and high energy physics (HEP) research communities.

Posting the PHENIX data to this database, known as "HEPData," would make it accessible to anyone wanting to compare new findings with historical measurements or results from one experiment to another—or see how experimental results match up with theoretical descriptions of the building blocks of matter.

Record from hepdata.net

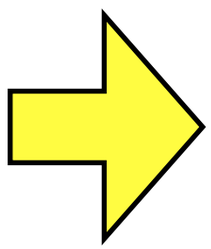


- DOIs minted for whole (versioned) record and each table.

Data output formats

hepdata.net/formats

YAML: native HEPData format.



`submission.yaml`
+ YAML data files for each table
+ optional resource files

- JSON: JavaScript Object Notation.
- CSV: comma-separated values.
- YODA: for inclusion in a Rivet analysis.
- ROOT: binary `.root` file.

`https://www.hepdata.net/record/ins1860766?format=csv`

- Programmatic access to download data in different formats.
- `format={original,yaml,json,csv,yoda,root}`
- Additional (optional) arguments for `version` and `table`.
- `curl` record w/o `format` returns Schema.org JSON-LD.

hepdata-cli

- CLI and Python API for HEPData search/download/upload.
- Summer project in 2020 by Giuseppe De Laurentis.
- Install (in venv) with: `pip install hepdata-cli`
- Examples of usage:

```
hepdata-cli find 'collaborations:"Belle-II"' -i inspire
```

```
hepdata-cli fetch-names 1860766 -i inspire
```

```
hepdata-cli download 1860766 -f csv -i inspire
```

```
hepdata-cli upload /path/to/TestHEPSubmission.tar.gz -e
my@email.com -p $PASSWORD –r 123456 -i $INVITATION_COOKIE -s False
```

Code: `https://github.com/HEPData/hepdata-cli`

`submission.yaml` file

- Separate YAML "document" for each data table, e.g.

```yaml
---
# This is Table 3.
name: "Table 3"
location: Data from Figure 8A
description: Normalized ZZ fiducial cross section (multiplied by 10^6 for
keywords: # used for searching, possibly multiple values for each keyword
- {name: reactions, values: [P P --> Z0 Z0 X]}
- {name: observables, values: [DSIG/DPT]}
- {name: cmenergies, values: [7000.0]}
- {name: phrases, values: [Inclusive, Single Differential Cross Section,
data_file: data3.yaml
additional_resources:
- {description: Image file, location: figFigure8A.png}
- {description: Thumbnail image file, location: thumb_figFigure8A.png}
```

- Validated against `submission_schema.json`

YAML data file

- "Independent" and "dependent" variables:

```yaml
independent_variables:
- header: {name: Leading dilepton PT, units: GEV}
  values:
  - {low: 0, high: 60}
  - {low: 60, high: 100}
  - {low: 100, high: 200}
  - {low: 200, high: 600}
dependent_variables:
- header: {name: 10**6 * 1/SIG(fiducial) * D(SIG(fiducial))/DPT, units: GEV**-1}
  qualifiers:
  - {name: RE, value: P P --> Z0 < LEPTON+ LEPTON- > Z0 < LEPTON+ LEPTON- > X}
  - {name: SQRT(S), units: GEV, value: 7000}
  values:
  - value: 7000
    errors:
    - {symerror: 1100, label: stat}
    - {symerror: 79, label: 'sys,detector'}
    - {symerror: 15, label: 'sys,background'}
  - value: 9800
    errors:
    - {symerror: 1600, label: stat}
    - {symerror: 75, label: 'sys,detector'}
    - {symerror: 15, label: 'sys,background'}
  - value: 1600
    errors:
    - {symerror: 490, label: stat}
    - {symerror: 41, label: 'sys,detector'}
    - {symerror: 2, label: 'sys,background'}
  - value: 80
    errors:
    - {symerror: 60, label: stat}
    - {symerror: 2, label: 'sys,detector'}
    - {symerror: 0, label: 'sys,background'}
```

RE	P P --> Z0 < LEPTON+ LEPTON- > Z0 < LEPTON+ LEPTON- > X
SQRT(S)	7000.0 GeV
Leading dilepton PT [GEV]	**10**6 * 1/SIG(fiducial) * D(SIG(fiducial))/DPT [GEV**-1]**
0.0 - 60.0	7000.0 **±1100.0 stat** **±79.0 sys,detector** **±15.0 sys,background**
60.0 - 100.0	9800.0 **±1600.0 stat** **±75.0 sys,detector** **±15.0 sys,background**
100.0 - 200.0	1600.0 **±490.0 stat** **±41.0 sys,detector** **±2.0 sys,background**
200.0 - 600.0	80.0 **±60.0 stat** **±2.0 sys,detector**

- Validated against `data_schema.json`

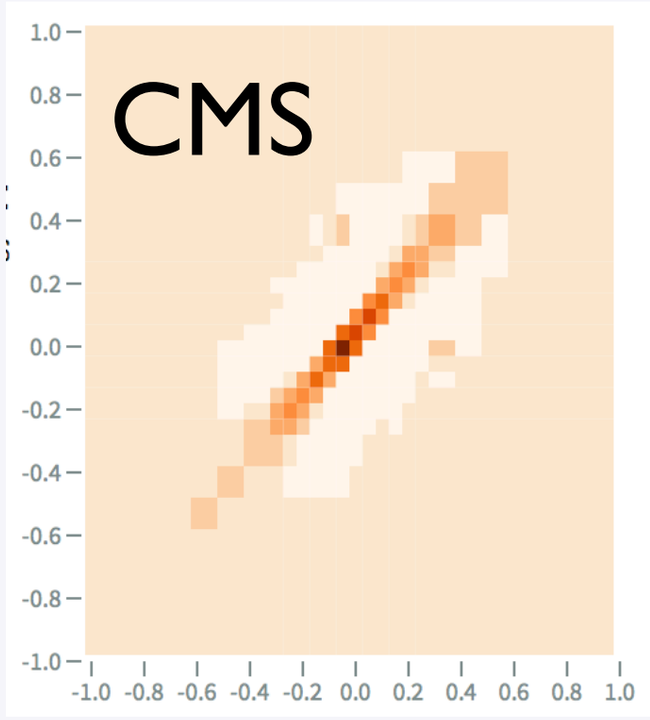
Covariance matrices

Correlation/covariance matrices hepdata-submission.readthedocs.io

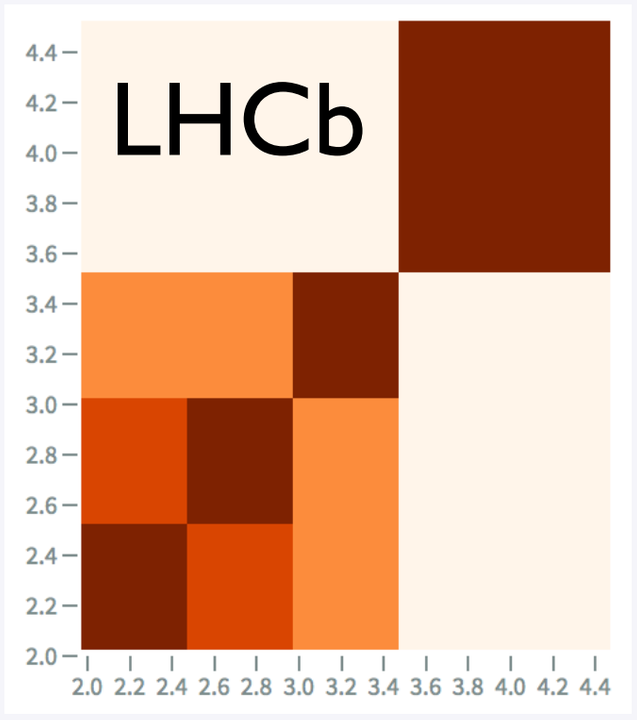
Correlation/covariance matrices can be encoded in a format with two independent variables (giving the bins) and one dependent variable (giving the covariance/correlation), e.g.

```yaml
independent_variables:
- header: {name: PTjet, units: GeV}
  values:
  - {low: 25, high: 45}
  - {low: 45, high: 65}
  ...
- header: {name: PTjet, units: GeV}
  values:
  - {low: 25, high: 45}
  - {low: 25, high: 45}
  - {low: 45, high: 65}
  ...
dependent_variables:
- header: {name: Correlation}
  values:
  - {value: 1.0000}
  - {value: 0.8727}
  - {value: 1.0000}
  ...
```

ins1605749



ins1454404

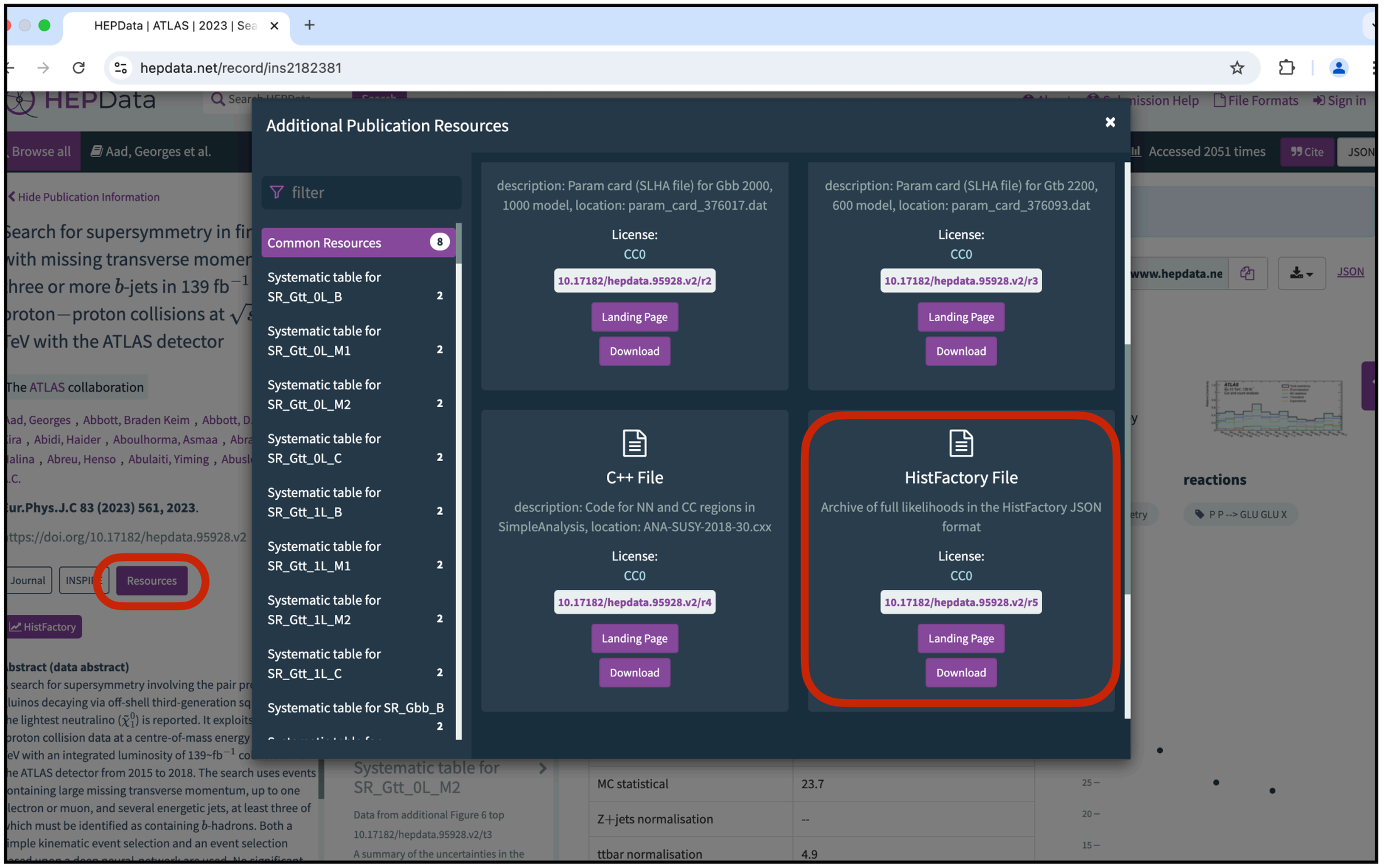


- Export to JSON, CSV, ROOT (TGraph2DErrors, TH2F), YODA (Estimate2D).

Submission documentation

- Documentation at hepdata-submission.readthedocs.io. Includes example Python scripts (simple, complicated).
- HEPData YAML files checked against JSON schema by validation code during submission. `hepdata-validate`
- hepdata_lib package by *Clemens Lange* (and *Andreas Albert)*. Library to read in text/ROOT and write HEPData YAML. https://github.com/HEPData/hepdata_lib
- Similar Python package by *Christian Holm Christensen*. https://gitlab.com/cholmcc/hepdata
- Experiments often develop internal HEPData tools/docs.

Additional resource files



- Search query `analysis:HistFactory` finds records.

NEW

from May 2024

Searching for resources

Searching resources by field

Text-based description searching:

resources:"Created with hepdata_lib"

Quotes force a full match.

Resource-type searching:

resources.type:png

Examples: png, html, github, zenodo etc.

Searching for specific URLs:

resources.url:atlas.web.cern.ch

- Additional resource metadata now indexed for searching.

Links to analysis code

- Rivet provides an interface between data and theory.
- JSON file maps INSPIRE IDs to Rivet analysis names:

```
{"100016": [“GAMMAGAMMA_1975_I100016”], …,
     “954993”: [“ATLAS_2011_I954993”]}
```

- Nightly task parses JSON file and adds new analyses.
- Search query `analysis:rivet` to find records.
- Extended to other analysis frameworks containing publication-specific code:
  - `analysis:MadAnalysis` (from Oct 2023)
  - NEW `analysis:SModelS` (from Nov 2024)

Follow @HEPData

Summary

Email: info@hepdata.net
Forum: hepdata-forum.cern.ch

- **HEPData** is *the* repository for HEP (and some nuclear) data.
- Widely used by HEP community: 4.5 million page views in 2023.
- *Caveats*: design restricts size (≲MB) and format (mostly tabular).
- Transition in 2017 to modern hepdata.net site hosted by CERN.
- Responsibility for data submission now held by experiments.
- **Future plans:**
  - Import data collections from Rivet and NUISANCE.
  - OpenMAPP to enhance interfaces with analysis toolkits.
  - Possible integration in European Open Science Cloud (EOSC).
  - Citation tracking of HEPData DOIs via INSPIRE-HEP.
- Interested in perspectives from the nuclear data community!