HEPData: a repository for high energy physics data

Lukas Heinrich[1], **Eamonn Maguire**[2] **and Graeme Watt**[3]

[1] Department of Physics, New York University, New York, USA
[2] CERN, Geneva, Switzerland
[3] IPPP, Department of Physics, Durham University, Durham, UK

E-mail: Graeme.Watt@durham.ac.uk

Abstract. The Durham High Energy Physics Database (HEPData) has been built up over the past four decades as a unique open-access repository for scattering data from experimental particle physics. It currently comprises data points underlying several thousand publications. Over the last two years, the HEPData software has been completely rewritten as an overlay on the Invenio v3 framework. The software is open source with the new site available at https://hepdata.net now replacing the previous site at http://hepdata.cedar.ac.uk. In this write-up, we describe the development of the new site and explain some of the advantages it offers over the previous platform.

1. Introduction

The Durham High Energy Physics Database (HEPData), a unique open-access repository for scattering data from experimental particle physics, has a long history dating back to the 1970s. It currently comprises data related to several thousand publications including those from the Large Hadron Collider (LHC). These are generally the numbers corresponding to the data points either plotted or tabulated in the publications, "Level 1" according to the DPHEP [1] classification, and HEPData is therefore complementary to the recent CERN Open Data Portal (http://opendata.cern.ch) which focuses on the release of data from Levels 2 and 3. The traditional focus of HEPData has been on measurements such as production cross sections and so the domain differs from the compilation of particle properties provided by the Particle Data Group (http://www-pdg.lbl.gov). In recent years HEPData has expanded beyond the traditional (unfolded and background-subtracted) measurements to also include data relevant for "recasting" LHC searches for physics beyond the Standard Model (BSM). The scope of HEPData is also being broadened to include data from particle decays and neutrino experiments, and potentially low-energy data relevant for tuning of the Geant4 detector simulation toolkit.

The HEPData project last underwent a major redevelopment around a decade ago [2], as part of the work of the CEDAR collaboration [3], where data was migrated from a legacy hierarchical database to a modern relational database (MySQL) and a web interface built on CGI scripts was replaced by a new Java-based web interface. The old HepData site (http://hepdata.cedar.ac.uk) ran on a single machine hosted at the Institute for Particle Physics Phenomenology (IPPP) at Durham University. Over the last two years, a complete rewrite has once again been undertaken to use more modern computing technologies. The new site (https://hepdata.net) is hosted on a number of machines provided by CERN OpenStack and offers several advantages and new features compared to the old site. In this write-up, we

describe the development of the new HEPData site (note the slight rebranding denoted by the change in capitalisation of "HEPData" compared to the old "HepData"). The code is open source and available from a dedicated GitHub organisation (`https://github.com/HEPData`).

2. Migration

The old HepData site stored all information in a (MySQL) database. To add a new record, all data needed to be first manually transformed into a standard "input" text format, consisting of metadata for each table followed by data points in a structured format. The uploaded input file was then parsed by a script to insert the information into the database. For the new HEPData site, we decided to store records as text files rather than in a database, since only the metadata (and not the actual data points) needs to be made searchable. Rather than retaining the old *ad hoc* "input" text format, we defined a new text format (`http://github.com/HEPData/hepdata-submission`) using YAML (`http://yaml.org`), which is similar to JSON but more human-readable. We investigated the possibility of a new universal input format using ROOT (`https://root.cern.ch`), but it was not suited to representing all of the diverse data types already present in HepData, particularly the metadata that describe each of the tables. All data from the existing HepData database was then exported to the new YAML format for migration to the new system. A validator (`http://github.com/HEPData/hepdata-validator`) was also written to ensure that the YAML files conform to the defined schema. Migrated YAML files (and future submitted files) are stored on the CERN EOS file system.

3. Software

The HEPData software was rewritten from the ground up, predominantly in the Python and JavaScript programming languages, as an overlay on the Invenio v3 (`http://inveniosoftware.org`) digital library framework, but with a very large degree of customisation. A screenshot of a typical data record is shown in figure 1. As with the previous HepData site, the main added value of storing the data in a standard format is that the data points can be automatically converted to various formats (see section 5) and visualised. Tables and scatter plots, or heatmap plots if there is more than one independent variable, are rendered with custom JavaScript code making use of the D3.js library (`https://d3js.org`), with options to switch on and off various elements. Auxiliary files, such as the original ATLAS plot shown in the top-right of figure 1, can be attached or linked to either individual tables or the whole record. A semi-automated way was developed to make links to analysis code within frameworks such as Rivet [4] (see middle panel of figure 1), where the framework authors provide a JSON file listing available analyses.

4. Discoverability

The new HEPData software uses a PostgreSQL database (with a new data model compared to the previous database), indexed with Elasticsearch to provide fast and powerful searching across all metadata fields. A screenshot of a typical search is shown in figure 2. More specific searches on keywords such as *cmenergies*, *observables*, *phrases* and *reactions* are also possible. Faceted search is also implemented for certain fields; see the left-hand panel in figure 2. All content in HEPData is semantically enriched using *Schema.org* vocabulary. This means that Google and other search engines know more about the content, and that the content can be automatically retrieved and interpreted by developers. An alternative data-driven search module for HEPData has been developed by Alicia Boya García for a Master's project at the University of Salamanca and a prototype is available at `http://hepdata.rufian.eu`.

In 2012, a first attempt was made to integrate the old HepData site with the Inspire HEP literature database (`http://inspirehep.net`) by harvesting the HepData tables and creating new Inspire records for each table. This first attempt was an important step forward, but the integration was never fully completed. The Inspire service is also being rewritten using

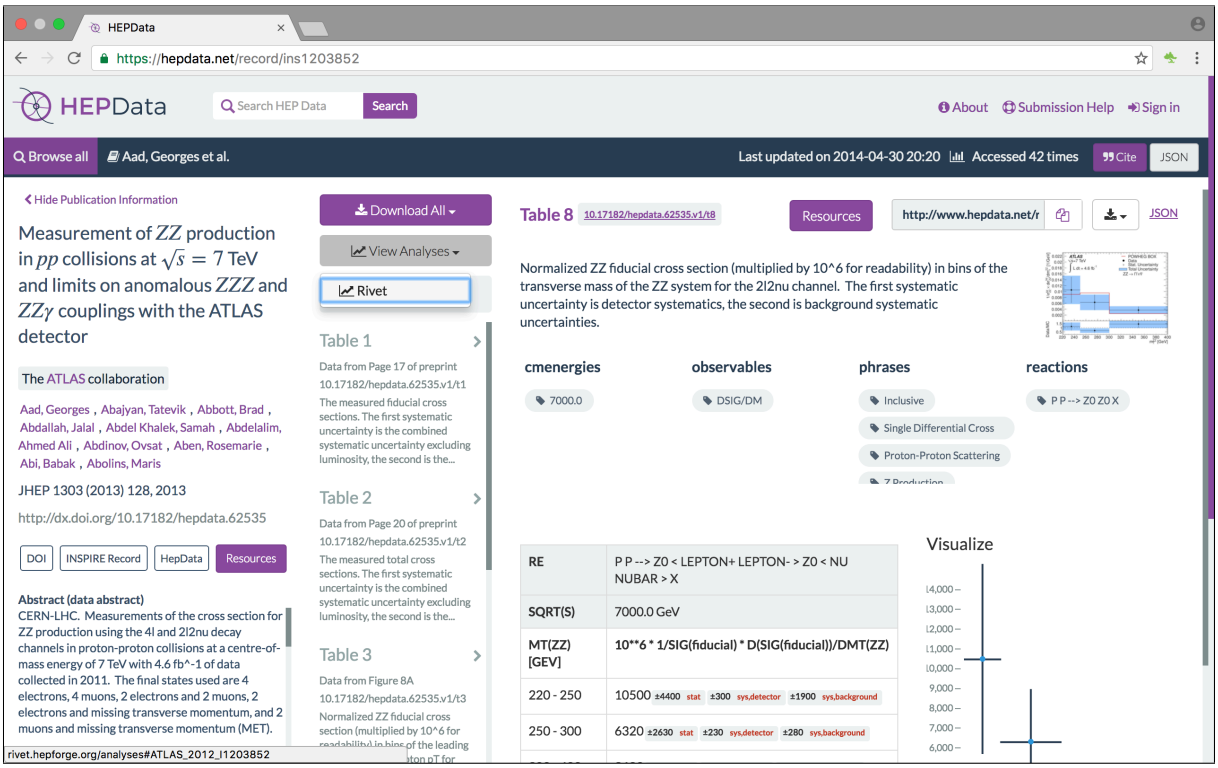

Figure 1. Screenshot of a typical record on the new HEPData site.

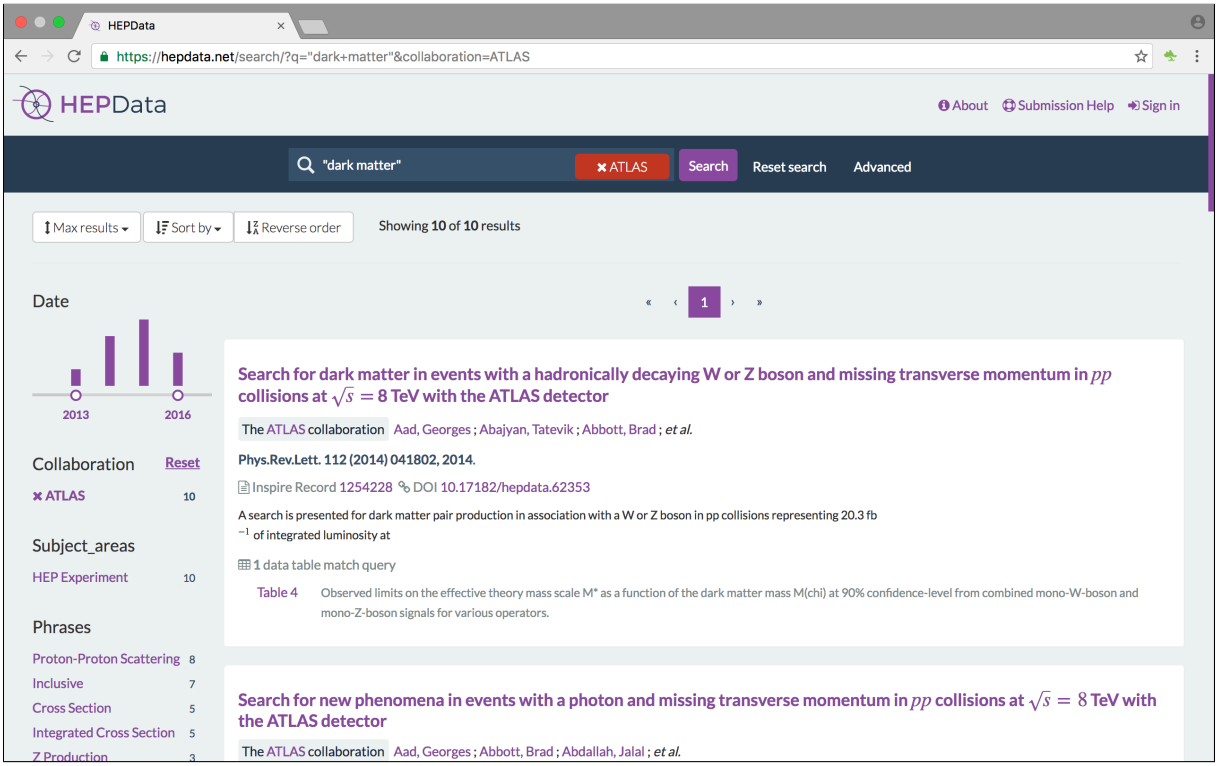

Figure 2. Screenshot of a typical search on the new HEPData site.

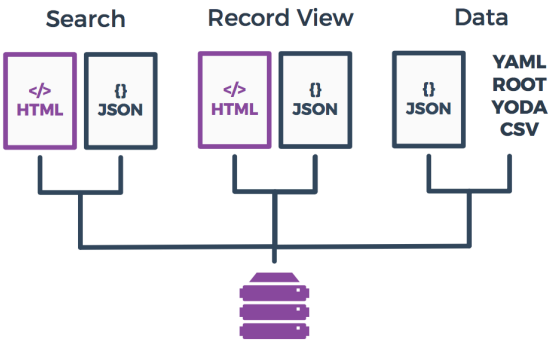

Figure 3. Diagram showing programmatic access and export formats. Search results, record display, and data tables can be obtained in a JSON format. Data tables can be exported in either the native YAML format, or converted to alternative CSV, ROOT and YODA formats.

Invenio v3 and so integration with the new HEPData site should be straightforward. The new HEPData records are now versioned in a way similar to arXiv papers, meaning that mistakes in an original submission can be corrected with a "version 2" and the original "version 1" is retained and not overwritten. Digital object identifiers (DOIs) are minted separately for each version via DataCite for both whole records and individual tables. Assignment of DOIs is made before a HEPData record has been finalised, meaning that the HEPData record DOI can be cited in the first arXiv version of the corresponding publication. Eventually, the DOIs will allow citation of HEPData records to be tracked by Inspire in a similar way to publications.

5. Conversion

The HTML web pages for search results and record display on the new HEPData site all have a JSON equivalent; see figure 3. The JSON format allows programmatic access, from applications such as Mathematica, simply by adding the option `format=json` to the URL. Conversion to various export formats is provided by a separate package (`http://github.com/HEPData/hepdata-converter`), originally developed by a CERN summer student, Michał Szostak [5]. Current output formats are listed below:

YAML The native HEPData format of the data tables (see `hepdata-submission`).

CSV A simple text format consisting of comma-separated values, which can be imported into many widely-used applications such as Excel.

ROOT A binary `.root` file rather than a CINT script, with each table in a separate directory. For numeric data, a `TGraphAsymmErrors` object is written for each dependent variable. If the data has finite bin widths, then also separate `TH1F` objects are written for the central value of the data points and each of the uncertainties. If there is more than one independent variable, the appropriate ROOT object (`TH2F` or `TH3F`) is chosen instead of a `TH1F` object.

YODA The data analysis classes used in the Rivet toolkit [4]. Again, the appropriate YODA object (`Scatter1D`, `Scatter2D`, `Scatter3D`) is written according to the number of independent variables in a table.

The HEPData DOIs are written to the various output formats to allow a user to later track the origin of the downloaded data points. A further package runs the converter as a web service, which is deployed inside a Docker container including dependencies such as the ROOT and YODA packages. Data can be accessed from a script via predictable URLs, for example,

`https://hepdata.net/record/ins1283842?format=yaml&table=Table1&version=1`

for `format={csv,json,root,yaml,yoda}`. Omitting the table name gives all tables (in a `.tar.gz` file unless JSON) and omitting the version number gives the latest version. An option `light=True` for the JSON format of a whole submission omits the data tables.

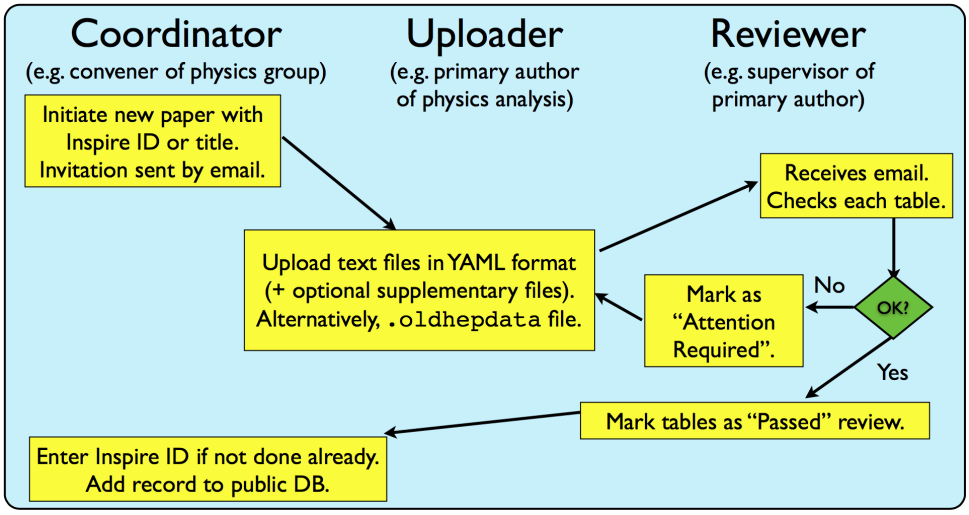

Figure 4. Submission flowchart for the new HEPData site.

6. Submission

The new HEPData submission flowchart is shown in figure 4. The procedure has evolved from the workflow successfully used for external submissions on the old HepData site for more than two years, with an average external submission rate of around 12 papers/month, primarily from the four main LHC experiments (ALICE, ATLAS, CMS, LHCb). We define three different roles:

Coordinator Typically a fairly senior person within an experimental collaboration, such as a convener of a physics group within ATLAS or CMS, who initiates the submission and is responsible for its final approval on behalf of the collaboration.

Uploader A primary author of a particular physics analysis, perhaps a Ph.D. student, who prepares and uploads the submission in one of the supported standard input formats.

Reviewer A more senior person familiar with the particular physics analysis, perhaps the supervisor of the primary author. The Coordinator can also choose to act as the Reviewer.

All current submissions are available for a Coordinator, Uploader, or Reviewer to see in their Dashboard (`https://hepdata.net/dashboard`). Prospective Coordinators should select the "Request Coordinator Privileges" option from their Dashboard and enter the name of the experiment/group that they wish to coordinate. A list of current Coordinators can be seen at `https://hepdata.net/permissions/coordinators`.

The Coordinator initiates a new submission by clicking the "Submit" button after logging in, then entering either the Inspire (`http://inspirehep.net`) record number of the corresponding publication (assigned automatically after the paper appears on the arXiv) or a provisional paper title. The Coordinator assigns an Uploader and Reviewer, then an email is sent to the designated Uploader with a link which gives them the privileges to upload a data submission. Once the Uploader thinks their submission is ready for review, they should click the "Notify Reviewer" button on the record. An email is then sent to the designated Reviewer with a link which assigns them the appropriate privileges. The Reviewer marks each table as either "Passed" review or "Attention Required" via a widget displayed next to each table, and messages can be entered to provide feedback to the Uploader. The Uploader can reply to these messages (which are also sent by email) via the widget before uploading a revised submission with corrections. Plots displayed automatically beside each table help to find mistakes in numerical data points. The Reviewer needs to mark each table as having "Passed" review before the Coordinator can

"Finalise" the submission from their Dashboard. The entire submission will then be published and made searchable in HEPData. It will also appear on the homepage under "Recently Updated Submissions" and a Tweet will be posted automatically to the @HEPData Twitter account. Further explanation of the submission steps can be found at `https://hepdata.net/submission` and in some independent documentation [6].

In contrast to the old HepData system, a submission can now be initiated and reviewed without knowing the Inspire record number of the corresponding paper. This means that a HEPData record can be prepared simultaneously with the corresponding paper. The HEPData record can then be finalised soon after the paper appears on the arXiv generating an Inspire entry. Paper metadata is now pulled directly from Inspire, therefore it does not need to be included explicitly in the data input file, as was the case with the old HepData system.

The primary submission format consists of a `submission.yaml` file containing metadata together with a YAML data file for each table containing the independent and dependent variables (see `http://github.com/HEPData/hepdata-submission`). In addition, there might be some auxiliary files associated with either the whole submission or individual tables. All these files should be uploaded in a single archive (`.zip`, `.tar`, `.tar.gz`). However, if there are no auxiliary files, the upload system also accepts a single YAML file containing both metadata and data. This format was used for migration and can be obtained by appending "`/yaml`" to any of the old HepData record URLs. To ease the transition to the YAML format, the new submission system also accepts the old "input" format in a single text file with extension `.oldhepdata`, which will be automatically converted by the `hepdata-converter` to the new YAML format.

A *Sandbox* (`https://www.hepdata.net/record/sandbox`) allows any logged-in user to upload a submission without any special permissions. A Sandbox record has a persistent URL (containing a 10-digit identifier) that can be shared to allow access by anyone. The Sandbox is therefore convenient for testing uploads and sharing access to records that should not be made searchable through the main HEPData site. A Sandbox record can be removed (by the user who created it) when it is no longer required.

7. Future plans

While HEPData has so far only been used for data associated with experimental particle physics papers, it could also be used to store numerical values of theoretical predictions and related material from particle physics phenomenology papers, without any necessary changes to the software or submission workflow. There is potential to store low-energy data from nuclear, atomic, and medical physics, relevant for validation of the Geant4 (`http://geant4.cern.ch`) toolkit, but further software development may first be needed to support keywords specific to the low-energy data and to support creation of records where the associated publications do not appear on Inspire.

In future we plan to support a mixed YAML/ROOT input format where metadata is provided in YAML files (as before), but numerical values are extracted from ROOT objects and converted to the standard YAML format. HistFactory [7] is a framework used for many ATLAS BSM (and Higgs) analyses, based on ROOT histograms organised by XML files which specify the different channels, the signal and background samples, and (correlated) systematic uncertainties associated with these samples. Some preliminary work has been done to extract HEPData tables in the standard YAML format from a HistFactory configuration. A more ambitious idea would be to define custom data types beyond a simple table, such that the HistFactory configuration could be stored directly without loss of information. A first step has been taken in this direction by allowing custom JSON schema to be specified in the `hepdata-validator` package.

8. Summary

The software underlying the Durham High Energy Physics database (HEPData) has been completely rewritten over the last two years, predominantly in the Python and JavaScript programming languages, as an overlay on the Invenio v3 digital library framework, but with a very large degree of customisation. The new site (`https://hepdata.net`) is now hosted at CERN on the OpenStack infrastructure, but still managed remotely from Durham. The transition from the old site (`http://hepdata.cedar.ac.uk`) has effectively been completed, with all data records being migrated to the new site. Future submissions should use the new site, which offers a variety of improvements compared to the old one.

In conclusion, the new HEPData site provides a state-of-the-art web platform for particle physicists to make their data *Findable*, *Accessible*, *Interoperable*, and *Reusable* according to the FAIR principles (see `https://www.force11.org/group/fairgroup/fairprinciples`).

Acknowledgments

HEPData is funded by a grant from the UK Science and Technology Facilities Council. We thank Mike Whalley for his dedicated 34 years of service as Database Manager for previous incarnations of the HEPData project, and for his help in migrating the data to the new platform.

References

[1] Mount R *et al.* (DPHEP Study Group) 2009 (*Preprint* `0912.0255`)
[2] Buckley A and Whalley M 2010 *PoS* **ACAT2010** 067 (*Preprint* `1006.0517`)
[3] Buckley A 2007 *PoS* **ACAT2007** 050 (*Preprint* `0708.2655`)
[4] Buckley A *et al.* (Rivet) 2013 *Comput. Phys. Commun.* **184** 2803–2819 (*Preprint* `1003.0694`)
[5] Szostak M F 2015 URL `https://cds.cern.ch/record/2055193`
[6] Bonanomi M and Marcoli M 2016 URL `https://doi.org/10.5281/zenodo.197109`
[7] Cranmer K *et al.* 2012 URL `https://cds.cern.ch/record/1456844`