Uploading data on HEPData

Matteo Bonanomi, Matteo Marcoli

HEPData

1 Introduction

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 the data points from plots and tables related to several thousand publications including those from the Large Hadron Collider (LHC). HEPData is funded by a grant from the UK STFC and is based at the IPPP at Durham University. Recently, thanks to a collaboration between the Durham University and INFN[1], this database has been extended to cross section for the ionization by electron impact data. The aim of this collaboration is to extend this database to any kind of physical data.

2 Requirements

In order to upload your data on the database, you need to respect some requirements. This document will guide you through the entire process for the upload of your data. To upload your data on HEPData it is required that they are stored into specific `.yaml` files. The creation of these files is automatized, starting from a `.txt` file where your data are stored, thanks to a `.txt` to `.yaml` converter, described in section 2.2.

2.1 Your file

The first step for the upload is the creation of a `.txt` file, where your data are stored. In order to make the converter work properly, this file must have a well defined structure. What you are going to import are couple of values, defined in the following as *dependent* and *independent variables*. Generally he former have an associated error, while the latter does not. Hence your file has to be structured in three columns, namely: independent variable, dependent variable and associated error. Since these variables can be any kind of physical data, you have to specify what do they represent and their unit of measurement. This has to be done in the first two rows of your `.txt` file. More precisely: the first row must contain (in this order) names for *independent variable, dependent variable* and *associated error*; the second row must contain unit of measurement for these three variables. It follows an example of right structure for your `.txt` file:

[1]Italian National Institute for Nuclear Physics

Energy	CS	Error
KEV	B	B
5.11E+01	1.49E+02	3.80E+01
5.65E+01	1.76E+02	3.20E+01
6.11E+01	1.88E+02	4.50E+01
6.59E+01	1.81E+02	5.30E+01

Note that names and typesettings chosen at this point will be fixed for the rest of the uploading process. That is: if you want "Energy" to be displayed as "E" on the site and in the plots that will be produced you have to name it as "E" instead of "Energy" in the first line of the `.txt` file. For what concerns unit of measurment and independent variables, you cannot set them arbitrairly but you have to follow HEPData conventions, as defined here. Further details about HEPData requirements and conventions can be found in section 2.3.

2.2 Txt to yaml converter

In order to upload your data on HEPData, you need them to be stored in particular `.yaml` files. For the creation of these files you can find appropriate schemes here. Since this procedure can be boring and stressing if done by hand, once you have created your `.txt` files containing your data sets, you can use a specific `.txt` to `.yaml` converter. This will generate all the `.yaml` files needed (almost) automatically. This converter is written in `cpp` and it is completely accessible from our GitHub repository. In the repository you will find two different source files: `txt2yaml.cpp` and `submission_generator.cpp`. First of all you need to download and compile them on your machine. To do so, simply use:

```
$ c++ -o txt2yaml txt2yaml.cpp
$ c++ -o submission_generator submission_generator.cpp
```

Once you have created the executable files you can proceed with the conversion. The first step consists in the creation of suitable `.yaml` files for data. Each set of data should be reported in a different `.yaml` file. To generate it just use:

```
$ ./txt2yaml Table.txt Table.yaml
```

with `data.txt` being your data file written according to the requirements exposed before and `data.yaml` the file you want to store your data in. You have to repeat this procedure for each set of data you want to appear in

your submission. When these files are ready, it is possible to generate the `submission.yaml` file. The generation requires a simple file with the names of `.yaml` files created for each set of data, one in each row:

```
Table1.yaml
Table2.yaml
Table3.yaml
...
```

Eventually, you create the `submission.yaml` file with:

```
$ ./submission_generator filenames.txt
```

where `filenames.txt` is the file containing the names of `.yaml` data files. Now the `submission.yaml` file has been created, but it is not complete yet. It has to be filled with proper tags and keywords for data you are submitting. However the keywords have to follow precise rules and conventions in order to be accepted by the HEPdata system. These requirements will be discussed in 2.3.

2.3 HEPData requirements for dataset

If you look at the `submission.yaml` file, you can see that each dataset should be associated with specific necessary keywords. Their names are already specified in the creation of the file, while the respective values have to be inserted by the user between square brackets, accordingly to the type of data one is trying to submit. The keyword named *reactions* requires as values one ore more reactions that describe the physical process data are referred to. All the admitted reactions you can insert are listed here. The keyword *observables* indicates what data actually represent. Admitted values can be found here. The keyword *cmenergies* describes the energy (or the range of energies) at which measurements are taken[2].

Once you completed the keywords properly, you should add in the `submission.yaml` a file the description for each set of data and their location in the article, for example:

```
location: Page 17 of preprint
description: The measured fiducial cross sections.
```

Doing this adjustments, be careful not to modify the syntax of the `submission.yaml` file, otherwise it would not be compiled later.

[2]An energy range is defined by the format $[E_{min};\ E_{max}]$, while different values of energies are defined by the format $[E_1, E_2, E_3, \dots]$

For any further information such as record ID, references or data license you can find complete examples here. You will also find how to add *qualifiers* to your data in order to better specify the experimental conditions.

2.4 Finalizing the submission process

When your `submission.yaml` file and all the `.yaml` files containing data are ready you have to create a compressed archive with them in it in order to submit data. It is recommended to test your submission through the Sandbox tool available in HEPData. You can find the instructions to use it in 3. Eventually the submission procedure is described in 4.

3 HEPData Sandbox

Once you have created your `.yaml` files as described in section 2 you may like to have a preview on how they would look on HEPData. The site gives you this possibility, thanks to a tool called *Sandbox*[3]. Sandbox is a sort of private directory where you can do all your tests. Since Sandbox and the submission process are unrelated you can use it to test different configurations of your data sets without any risk. In order to upload your data on Sandbox it is required that they are stored into `.yaml` files and that all of them are stored in a `.tar.gz` or `.zip` directory, as described in section 2.3. Once you have compressed all your files in the same directory you are ready to upload them on Sandbox to see a first draft.

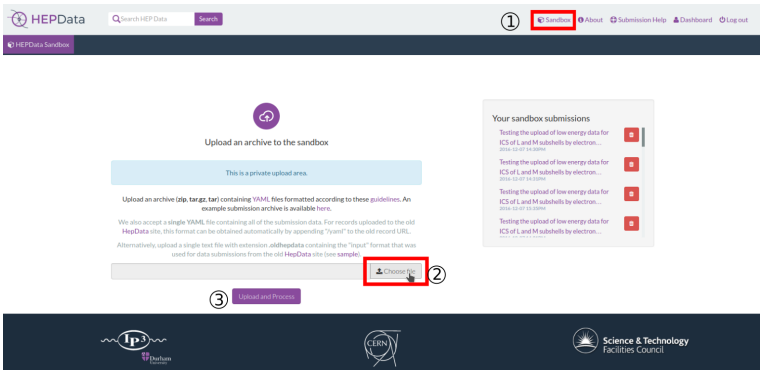

Figure 1: Sandbox view for the upload of a .tar.gz or .zip data set.

As said above, what you upload on Sandbox is not what you will submit at the final stage of your work. That is why you can also upload on Sandbox

[3]Accessible only when you are logged in

data that are not compliant with HEPData requirements described in 2.3. More precisely: on Sandbox you could define *keywords* and *reactions* not allowed in the production site without any particular problem. It is highly suggested to follow conventions and use standards defined in 2. Doing so the upload process will be quicker and simpler.

If you look at any data already present in HEPData and compare it with what you have on Sandbox you will note that there is a difference: the former has a title on the left, while the second does not. This is not a problem related to your `submission.yaml` file but it is a specific feature on HEPData. During the submission process you will be asked for an *inspire* identifier of the article linked to the data you are submitting: the title of this article will be the title of your dataset, once it is uploaded on the site. In the next section is presented the submission process in details.

4 The submission process

Once you have created your data archive as described in 2 and you have made all your tests on Sandbox, you are ready to upload your data on HEPData. The submission (and approval) of your data is a process involving three people, with three different roles: *uploader*, *reviewer* and *coordinator*.

4.1 Prerequisites for the upload

Ok, you followed the guide so far and now you want to put your data on HEPData. How to do that?

If you followed the guide properly now you should have your `submission.yaml` file and your data stored into suitable `.yaml` files[4].

If you tested the upload of your data on Sandbox, as described in section 3, you should also already have a `.tar.gz` or `.zip` archive of your `.yaml` files. If you skipped the Sandbox phase, before the upload, you need to create a compressed archive of all your `.yaml` files.

The last prerequisite for the upload is an *inspire* identifier of the article linked to the data you are submitting: at the moment being this is mandatory and can not be substitute with any other record. Hence you have to be sure that the article containing the data you want to submit is linked on inspire.

[4]If you did everything properly, each dataset should be stored in .yaml files called Table1.yaml, Table2.yaml and so on.

4.2 The upload process

Once you satisfied all the prerequisites, you can start the upload process. To do so, the first step is to get in touch with one of the HEPData *coordinators*. This person will follow the entire submission process and is the one who officially accepts your archive on the production site.

Once you have contacted a coordinator, you will receive an email as soon as your upload process has started and you will be asked for the inspire identifier of your article. Once you completed successfully this phase, you can verify it from your dashboard.

At this point of the process your article is present on the site in a draft mode. A *reviewer*, assigned to your upload request by the *coordinator*, will take care of verifying all your tables. You will receive an email if the reviewer comments one of your tables due to any kind of problem so that you can fix it. Generally you can encounter problems if you did not follow HEPData requirements, presented in section 2.

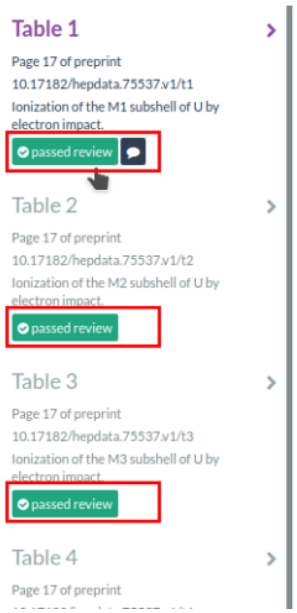

Figure 2: Example of tables after the review. The *balloon icon* in the first table means that comments have been done during the review phase.

Once the reviewer has approved all your tables, the coordinator needs to approve completely your archive and to complete the upload process. You will be notified by email once this last step has been completed.

Eventually you will find your data set on HEPData!