Uploading data to the HepDATA repository with Python using `hepdata_lib`

S. E. Müller

Helmholtz-Zentrum Dresden-Rossendorf

STRONG2020 Meeting on the Hadronic Cross Section database, December 18, 2020

Introduction

In the June 2020 meeting it has been agreed that the **HEPData** Repository shall be used to provide the input data for the **PrecisionSM** webpage.
This requires that the data should be uploaded to **HEPData** in a complete and consistent way.

- including statistical and systematic uncertainties and covariance matrices
- currently, this information is often scattered over several sources
  - original publication
  - supplemental material
  - webpage
  - additional document
- it would be very beneficial to have everything in one place in the **HEPData** repository

At the last meeting, I have presented the (sandbox) submission of the **KLOE10** data. Today, I'll try to provide a walk-through to the submission preparation.

- I have put everything into a **gitlab** project - please send your **github** -account to **stefan.mueller@hzdr.de** if you'd like to have access.

Prerequisites

I did prepare the submission on my Linux desktop running Ubuntu 18.04. This is what is needed:

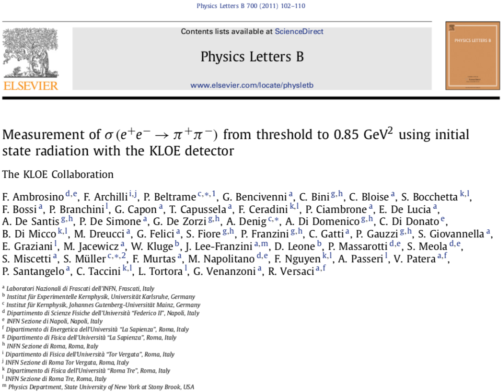
- **PYTHON**
  - probably already installed in your system
  - version 2 or 3 should both be ok (this depends what the **ROOT** installation was built with)
- the `hepdata_lib`
  - install with `pip install hepdata_lib`
  - it is recomended to do this in a virtual python environment
- the `virtualenv` python package for the virtual environment
  - install with your packagemanager (e.g. `apt install virtualenv`)
  - or using `pip` (needs to be installed):

```
pip install virtualenv
```

- **ROOT** with python support enabled
  - needs to be configured with support for python during installation (e.g. `-Dpython=ON` or `-Dpyroot=ON -DPYTHON_EXECUTABLE=/usr/bin/python3`
  - Do ’`root -config`’ and check output for `PYTHON_LIBRARY` to find out with which version **ROOT** has been built on your system

The KLOE10 publication

The KLOE10 data was originally published in
Phys. Lett. B700 (2011) 102 - 110:



Physics Letters B 700 (2011) 102–110

Contents lists available at ScienceDirect

Physics Letters B

www.elsevier.com/locate/physletb

Measurement of $\sigma(e^+e^- \to \pi^+\pi^-)$ from threshold to 0.85 GeV^2 using initial state radiation with the KLOE detector

The KLOE Collaboration

F. Ambrosino[d,e], F. Archilli[i,j], P. Beltrame[c,*,1], G. Bencivenni[a], C. Bini[g,h], C. Bloise[a], S. Bocchetta[k,l], F. Bossi[a], P. Branchini[l], G. Capon[a], T. Capussela[a], F. Ceradini[k,l], P. Ciambrone[a], E. De Lucia[a], A. De Santis[g,h], P. De Simone[a], G. De Zorzi[g,h], A. Denig[c,*], A. Di Domenico[g,h], C. Di Donato[e], B. Di Micco[k,l], M. Dreucci[a], G. Felici[a], S. Fiore[g,h], P. Franzini[g,h], C. Gatti[a], P. Gauzzi[g,h], S. Giovannella[a], E. Graziani[l], M. Jacewicz[a], W. Kluge[b], J. Lee-Franzini[a,m], D. Leone[b], P. Massarotti[d,e], S. Meola[d,e], S. Miscetti[a], S. Müller[c,*,2], F. Murtas[a], M. Napolitano[d,e], F. Nguyen[k,l], A. Passeri[l], V. Patera[a,f], P. Santangelo[a], C. Taccini[k,l], L. Tortora[l], G. Venanzoni[a], R. Versaci[a,f]

[a] *Laboratori Nazionali di Frascati dell'INFN, Frascati, Italy*
[b] *Institut für Experimentelle Kernphysik, Universität Karlsruhe, Germany*
[c] *Institut für Kernphysik, Johannes Gutenberg-Universität Mainz, Germany*
[d] *Dipartimento di Scienze Fisiche dell'Università "Federico II", Napoli, Italy*
[e] *INFN Sezione di Napoli, Napoli, Italy*
[f] *Dipartimento di Energetica dell'Università "La Sapienza", Roma, Italy*
[g] *Dipartimento di Fisica dell'Università "La Sapienza", Roma, Italy*
[h] *INFN Sezione di Roma, Roma, Italy*
[i] *Dipartimento di Fisica dell'Università "Tor Vergata", Roma, Italy*
[j] *INFN Sezione di Roma Tor Vergata, Roma, Italy*
[k] *Dipartimento di Fisica dell'Università "Roma Tre", Roma, Italy*
[l] *INFN Sezione di Roma Tre, Roma, Italy*
[m] *Physics Department, State University of New York at Stony Brook, USA*

Additional information was given in

- the KLOE10 webpage (for covariance matrices)
- the KLOE note 225 for systematic uncertainties

Submission script: Include files

We need to import hepdata_lib, numpy and, if you want to import data from **ROOT** histograms, the ROOT package:

```
#!/usr/bin/python

import hepdata_lib
from hepdata_lib import Submission
from hepdata_lib import Table
from hepdata_lib import Variable, Uncertainty
from hepdata_lib import RootFileReader
from hepdata_lib import root_utils

import numpy as np
import ROOT as r
```

Submission script: Data table preparation

I'd like to prepare the data for $\sigma_{\pi\pi}^{\text{bare}}$ from Phys. Lett. B700 (2011) 102 - 110. This information is found in Fig. 3, right and Table 2.
We need to create a `Table` object, give some description (the figure caption) and the location of the data in the paper:

```
table3b = Table("Figure 3b")
table3b.description = "Bare cross section for $e^+e^-\\rightarrow\pi^+\pi^-$"
table3b.location = "Data from Fig. 3, Right and Table 2"
```

Submission script: Data table preparation

I'd like to prepare the data for $\sigma_{\pi\pi}^{\text{bare}}$ from Phys. Lett. B700 (2011) 102 - 110. This information is found in Fig. 3, right and Table 2.
We need to create a `Table` object, give some description (the figure caption) and the location of the data in the paper:

```
table3b = Table("Figure 3b")
table3b.description = "Bare cross section for $e^+e^-\\rightarrow\pi^+\pi^-$"
table3b.location = "Data from Fig. 3, Right and Table 2"
```

We add information on predefined **HEPData** keywords for reactions, Observables and phrases:

```
table3b.keywords["reactions"] = ["E+ E- --> PI+ PI-"]
table3b.keywords["observables"] = ["SIG"]
table3b.keywords["phrases"] = ["Exclusive","E+E- Scattering",
                               "Integrated Cross Section","Cross Section"]
```

Submission script: Data table preparation

I'd like to prepare the data for $\sigma_{\pi\pi}^{\text{bare}}$ from Phys. Lett. B700 (2011) 102 - 110. This information is found in Fig. 3, right and Table 2.
We need to create a `Table` object, give some description (the figure caption) and the location of the data in the paper:

```
table3b = Table("Figure 3b")
table3b.description = "Bare cross section for $e^+e^-\\rightarrow\pi^+\pi^-$"
table3b.location = "Data from Fig. 3, Right and Table 2"
```

We add information on predefined **HEPData** keywords for reactions, Observables and phrases:

```
table3b.keywords["reactions"] = ["E+ E- --> PI+ PI-"]
table3b.keywords["observables"] = ["SIG"]
table3b.keywords["phrases"] = ["Exclusive","E+E- Scattering",
                               "Integrated Cross Section","Cross Section"]
```

We can also add the original figure from the paper:

```
table3b.add_image("./spp09.eps")
```

(make sure the "ImageMagick" package is installed on your system)

Submission script: Data table from text file

Define a `data3b` numpy array and load data from the `spp10.dat` file:

```
data3b = np.loadtxt("spp10.dat", skiprows=0)
```

Then create variables for $M_{\pi\pi}^2$, $\sigma_{\pi\pi}$, stat. and combined syst. uncertainty, fill them with corresponding array values:

```
mpp2_3b = Variable("$M_{\pi\pi}^2$", is_independent=True, is_binned=False,
units="GeV$^2$")
mpp2_3b.values = data3b[:,0]

spp3b = Variable("$\sigma_{\pi\pi}$",  is_independent=False, is_binned=False,
units="nb")
spp3b.values = data3b[:,1]

stat3b = Uncertainty("stat", is_symmetric=True)
stat3b.values = data3b[:,2]
spp3b.add_uncertainty(stat3b)

syst3b = Uncertainty("syst", is_symmetric=True)
syst3b.values = data3b[:,3]
spp3b.add_uncertainty(syst3b)
```

Submission script: Data table from text file (2)

Add some additional `qualifier` information to $\sigma_{\pi\pi}$ and add $M^2_{\pi\pi}$ and $\sigma_{\pi\pi}$ variable to the `table3b` object defined above:

```
spp3b.add_qualifier("SQRTS(S)",1000,"MeV")
spp3b.add_qualifier("RE","E+ E- --> PI+ PI-")

table3b.add_variable(mpp2_3b)
table3b.add_variable(spp3b)
```

Submission script: Data table from text file (2)

Add some additional `qualifier` information to $\sigma_{\pi\pi}$ and add $M^2_{\pi\pi}$ and $\sigma_{\pi\pi}$ variable to the `table3b` object defined above:

```
spp3b.add_qualifier("SQRTS(S)",1000,"MeV")
spp3b.add_qualifier("RE","E+ E- --> PI+ PI-")

table3b.add_variable(mpp2_3b)
table3b.add_variable(spp3b)
```

In the same way, read the data for the (16) individual contributions to systematic uncertainty:

```
table7_spp = Table("Table 7 spp")
...
data7_spp = np.loadtxt("create_syst/spp10_syst.dat", skiprows=0)
...
RcnFil_spp = Variable("Rec. Filter",is_independent=False, is_binned=False,
units="nb")
RcnFil_spp.values = data7_spp[:,1]

table7_spp.add_variable(mpp2_7_spp)
table7_spp.add_variable(RcnFil_spp)
table7_spp.add_variable(Bkgd_spp)
...
```

Submission script: Data table from ROOT file

Data can be also extracted from ROOT files:

```
reader = RootFileReader("histos.root")
cov_spp = reader.read_hist_2d("cov_spp")

covdata_spp = root_utils.get_hist_2d_points(cov_spp)

xcov_spp = Variable("Bin i $M_{\pi\pi}^2$ [GeV$^2$]", is_independent=True,
                     is_binned=True)
xcov_spp.values = covdata_spp["x_edges"]

ycov_spp = Variable("Bin j $M_{\pi\pi}^2$ [GeV$^2$]", is_independent=True,
                     is_binned=True)
ycov_spp.values = covdata_spp["y_edges"]

zcov_spp = Variable("Covariance values for $\\sigma_{\\pi\\pi}$",
                     is_independent=False, is_binned=False)
zcov_spp.values = covdata_spp["z"]
```

Submission script: Data table from ROOT file (2)

Create a `Table` object and add the variables `xcov_spp,ycov_spp,zcov_spp` to it:

```
tablecov_spp = Table("Covariance matrix values spp")
tablecov_spp.description = "Statistical covariance matrix for bare cross
                            section for $e^+e^-\\rightarrow\pi^+\pi^-$"
tablecov_spp.location = "Data from
                     https://www.lnf.infn.it/kloe/ppg/ppg_2010/ppg_2010.html"
tablecov_spp.keywords["observables"] = ["SIG"]
tablecov_spp.keywords["reactions"] = ["E+ E- --> PI+ PI-"]
tablecov_spp.keywords["phrases"] = ["Exclusive","E+E- Scattering",
                       "Integrated Cross Section","Cross Section"

for var in [xcov_spp,ycov_spp,zcov_spp]:
    tablecov_spp.add_variable(var)
```

Submission script: The “submission” object

Once we have all the data tables prepared, we need to create a `Submission` object:

```
K10submission = Submission()
```

Load an abstract from a text file:

```
K10submission.read_abstract("abstract_KLOE10.txt")
```

(Text file can contain LaTeXexpressions)

Add links and INSPIRE record:

```
K10submission.add_link("Webpage with all data files",
            "https://www.lnf.infn.it/kloe/ppg/ppg_2010/ppg_2010.html")
K10submission.add_link("Additional documentation KLOE Note 255 (June 2011)",
            "http://www.lnf.infn.it/kloe/kdocs/getfile.php?doc_fname=kn225.pdf")
K10submission.add_link("Additional documentation KLOE Note 255 (June 2011)",
             "https://www.lnf.infn.it/kloe/ppg/ppg_2010/kn225.pdf")
K10submission.add_link("arXiv", "https://arxiv.org/abs/1006.5313")
K10submission.add_record_id(859660, "inspire")
```

We can define an output directory to which the submission file will be written:

```
outdir = "KLOE10_HEPdata"
K10submission.create_files(outdir)
```

Submission script: The “submission” object (2)

Now we need to add all the tables to the `Submission` object:

```
K10submission.add_table(table3b)
K10submission.add_table(tablecov_spp)
K10submission.add_table(tableicov_spp)
K10submission.add_table(table7_spp)
```

We can assign the keyword for CM-energy of 1 GeV to all of the tables:

```
for table in K10submission.tables:
    table.keywords["cmenergies"] = [1.000]
```

Running the script

Once the python script is ready, it can be executed within a virtual environment using the `virtualenv` package. To create the virtual environment, do

```
virtualenv -p /usr/bin/python workdir/
source workdir/bin/activate
```

This will create a directory `workdir` with a virtual `python` environment inside and activates it. The `-p` flag allows to chose the desired `python` version (this should match the one needed by `ROOT`). `cd` into the directory and copy the submission script, all the data files, the abstract and the figures into it.
Then we load `hepdata_lib` by doing

```
pip install hepdata_lib
```

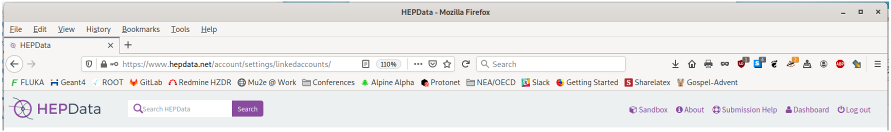
We execute the submission script:

```
python Submit_KLOE10.py
```

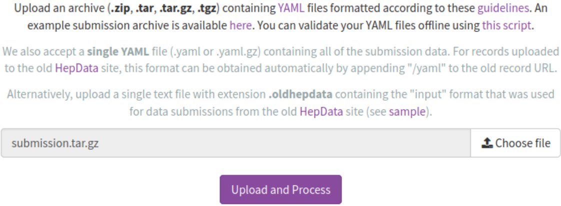
If all goes well, we should obtain a file `submission.tar.gz` in the specified output directory, which we can upload to a 'sandbox' on the **HEPData** webpage.
The virtual python session can be closed with the `deactivate` command (`hepdata_lib` will still be there at the next activation).

Uploading the submission to HEPData sandbox

Sign in to the **HEPData.net** page (e.g. with your ORCID account). Then click on “Sandbox” in the upper right:


HEPData - Mozilla Firefox
File Edit View History Bookmarks Tools Help
HEPData
https://www.hepdata.net/account/settings/linkedaccounts/
110%
Search
FLUKA Geant4 ROOT GitLab Redmine HZDR Mu2e @ Work Conferences Alpine Alpha Protonet NEA/OECD Slack Getting Started Sharelatex Gospel-Advent
HEPData
Search HEPData
Search
Sandbox About Submission Help Dashboard Log out

Select your `submission.tar.gz` in the “Choose file” menu, and click “Upload and Process:”



Upload an archive (**.zip**, **.tar**, **.tar.gz**, **.tgz**) containing YAML files formatted according to these guidelines. An example submission archive is available here. You can validate your YAML files offline using this script.

We also accept a **single YAML** file (.yaml or .yaml.gz) containing all of the submission data. For records uploaded to the old HepData site, this format can be obtained automatically by appending "/yaml" to the old record URL.

Alternatively, upload a single text file with extension **.oldhepdata** containing the "input" format that was used for data submissions from the old HepData site (see sample).

Your submission file will be validated, and you’ll receive an email about the status of your submission.

Summary

- The **HEPData** Repository shall be used to provide the input data for the **PrecisionSM** webpage
- It is desirable to have as many data sets as possible of hadronic cross section data uploaded to the **HEPData** repository
  - including correlation matrices and systematic uncertainties
- I tried to describe how I processed the KLOE10 data with the `hepdata_lib` python package for **HEPData** upload
  - more documentation can be found at the hepdata_lib examples
- I have not tried yet the *commandline-interface* (maybe Alberto has)
- What about validation tools for reviewers to ensure the uploaded data sets are correct? In `ROOT`?