

## PyPWA Manual - Fitting

**Step-by-step guide to PWA (mass-independent data fitting) using PyPWA**  
(using the scientific JLab batch farm)  
(February 2015)

J. Pond and C. Salgado  
Norfolk State University  
and  
The Thomas Jefferson National Accelerator Facility

All software used by PyPWA can be downloaded from <https://pypwa.jlab.org/>.  
Installing PyPWA software needs full internet access i.e go to jlabs1 as ifarm1102  
doesn't have internet access.

Create a PyPWA directory in your home directory.

Untar your software.

The general procedure of using PyPWA in the standard format (helicity formalism  
and isobar model) to perform partial wave analysis is the following:

**NOTE:** *It is advised that you read the general documentation in PyPWA/docs/, in the  
wiki PyPWA, or the website before proceedings. For the general formalism consult:  
C. W. Salgado and D. P. Weygand, Physics Report 537 (2014) 1-58 and references  
within.*

These are detailed step-by-step instructions for PWA (fitting):

### Requirements (You'll need your own software for these steps):

1) Analyze your data to select the signal and crate a gamp format file with all  
your events and name it **data\_events.gamp**

We also allow for the use of a Q factor (i.e. the probability for each event to be  
a signal, ie.  $Q = \text{signal} / (\text{signal} + \text{background})$ ), to be included in the PWA fit.  
Just create a file named **QFactor.txt** with the Q values of all events, one per  
line, written in the same order that the events are entered in data\_events.gamp.

2) Run a full monte carlo simulation (generate+geant(detector + reconstruction +  
analysis) using a flat phase-space generator (you can or can't include your t  
distribution there).  
Create two gamp formatted files: **raw\_events.gamp** with all the generated events,  
and **acc\_events.gamp** with all the events obtained after the full simulation.

You are ready to start using the PyPWA software.

**[1]** Log in to a CUE machine (e.g. ifarm1102). (Note: You may need to contact your

hall's scientific computing liason to get access to the Jlab scientific computing farm!)

**[2]** Create a directory named as you like (for example after your reaction: e.g. Pippimpi0 – called here “MAIN”)

**[3]** Move your files: data\_events.gamp, raw\_events.gamp, acc\_events.gamp and QFactor.txt(if you have one) into that directory.  
Copy PyPWA/pythonPWA/ from your home directory into that directory.

Go to the pythonPWA/batchFarmServices directory and

**[4]** run:

### fitting\_Install

A GUI called pwa\_controls will pop-up (Note: There is a "help" button in the GUI itself).

**[5]** Fill in the information and save.

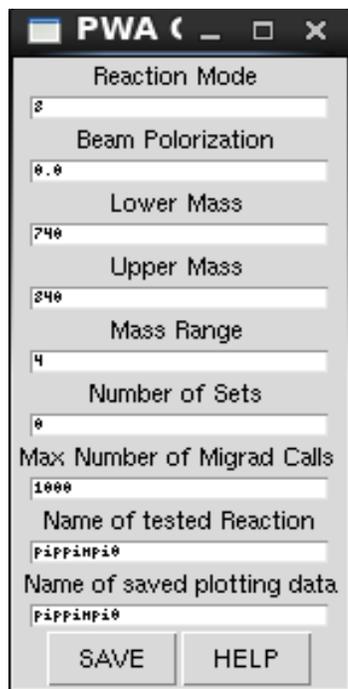
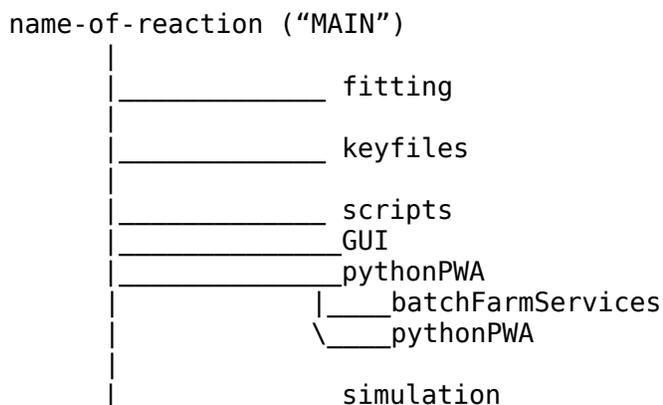


Figure 1: “Pwa\_controls” GUI. The HELP button helps you to navigate this GUI.

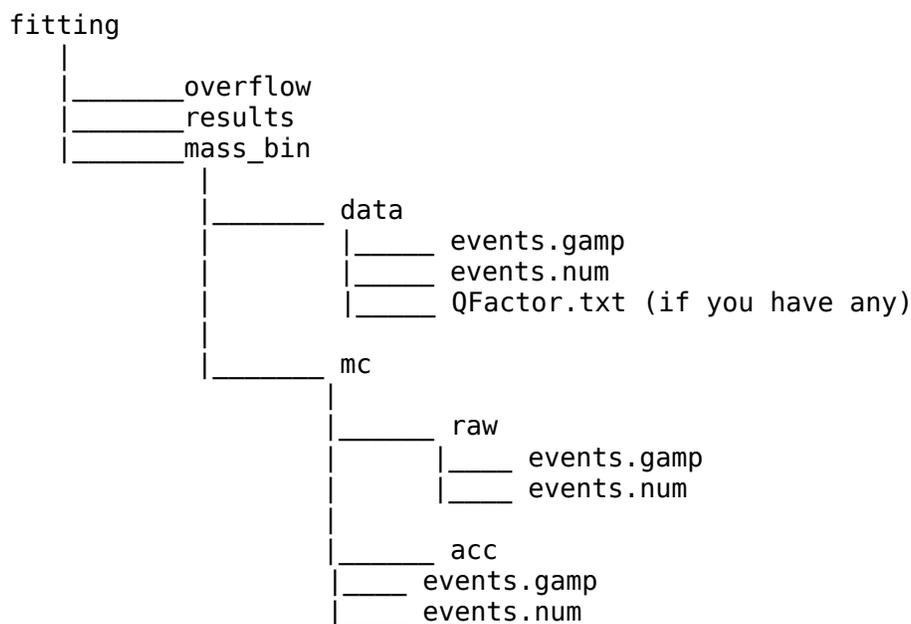
This action will create your full directory structure needed for your PWA. It can take up to 30 minutes of execution (if you have a lot of events). This action prepares the directory structure, re-bin the data, move data to the right

directories, transfer some information into numpy format and setup the necessary software.

You will have the following directory structure such that



and



-----

**[6]** Determine all waves you want to use and write a *name*.keyfile for each according to the gamp format (see general documentation). Populate the MAIN/keyfiles/ directory with all your waves as for example: 0-1--1-P\_rho.keyfile (*name* includes wave definition: IGJPCMepsL\_(isobar-if any).keyfile)

After this is done you are ready to start running farm jobs.

---

**[7]** go to the MAIN/GUI directory and run

### **PWA\_GUI**

This is the main GUI for the analysis. It will start with one column and after you make selections two new columns will be opening.



Figure 2: Main PWA GUI.

Then, what you need to do (in this order) is:

**[8]** click **fitting**

A new layer of buttons will open (second column in figure 3), (the next commands will send farm jobs to run the programs: *gamp*, *genAlpha* for each of the waves and data sets.

**[9]** click **Run Gamp**

**[10]** click **data**

**[11]** click **accMC**

**[12]** click **rawMC**

(these actions will create the necessary “waves” in binary format in each correct directory – files called *name.bamp*). These jobs can be run simultaneously.



Figure 3: Main GUI with selections for the second column and third column.

After clicking data, accMC and rawMC, command lines will be printed out (a line for each job, one per mass bin and keyfile) as they are being submitted to the batch system. All jobs, ( $3 * \# \text{ mass bins} * \# \text{ of waves}$ ) will all run as separate jobs and not interfere with each other.

**[13]** click **Gen Alpha**

**[14]** click **data**

**[15]** click **accMC**

**[16]** click **rawMC**

(these actions create the *alphaevents.txt* files in each directory)

You can submit the Run Gamp and Gen Alpha simultaneously, they will not interfere with each other.

**WAIT** until everything is done in the farm.

You need to look at <http://scicomp.jlab.org/scicomp/> to check that your jobs are all done and that there were successful (and with Exit Code of 0).

-----

**[17]** go **back** and click **normint**  
                           click **accMC**  
                           click **rawMC**

(these actions calculates accepted and raw normalization integrals for each mass bin)

You can run accMC and rawMC jobs simultaneously.

**WAIT** until everything is done in the farm.

You need to look at <http://scicomp.jlab.org/scicomp/> to check that your jobs are all done and that there were successful (and with Exit Code of 0).

-----

NOTE: before proceeding, check that all directories are filled with the necessary files for your PWA.  
The number of events in the files *alphaevents.txt*, *events.gamp* and *events.num* must be the same and the structure should look like this below (for example, for 9 waves in a 1000\_MeV mass bin):

```
1000_MeV/
|-- data
|   |-- 0++0-S.bamp
|   |-- 1--0-P.bamp
|   |-- 1--1-P.bamp
|   |-- 1--1+P.bamp
|   |-- 2++0-D.bamp
|   |-- 2++1-D.bamp
|   |-- 2++1+D.bamp
|   |-- 2++2-D.bamp
|   |-- 2++2+D.bamp
|   |-- alphaevents.txt
|   |-- events.gamp
|   |-- events.num
|   |-- events.npy
|   |-- rhoAA.npy
|   |-- QFactor.txt (if you have any)
|-- mc
|   |-- acc
|   |   |-- 0++0-S.bamp
|   |   |-- 1--0-P.bamp
|   |   |-- 1--1-P.bamp
|   |   |-- 1--1+P.bamp
|   |   |-- 2++0-D.bamp
|   |   |-- 2++1-D.bamp
|   |   |-- 2++1+D.bamp
|   |   |-- 2++2-D.bamp
|   |   |-- 2++2+D.bamp
|   |   |-- alphaevents.txt
|   |   |-- events.gamp
|   |   |-- events.npy
|   |   |-- events.num
|   |   |-- normint.npy
|   |-- raw
|   |   |-- 0++0-S.bamp
|   |   |-- 1--0-P.bamp
|   |   |-- 1--1-P.bamp
|   |   |-- 1--1+P.bamp
|   |   |-- 2++0-D.bamp
```

```
| -- 2++1-D.bamp
| -- 2++1+D.bamp
| -- 2++2-D.bamp
| -- 2++2+D.bamp
| -- alphaevents.txt
| -- events.gamp
| -- events.npy
| -- events.num
| -- normint.npy
```

---

-- [Fitting for each mass\\_bin \(Mass-independent fit using Minuit\)](#)

from the main GUI:

**[18]** *click Fitter* (second column after clicking FITTING)

All fitting jobs (one for each mass bin) will be submitted to the farm. The time required to run each job depends on number of events and number of waves used.

**WAIT** until everything is done in the farm.

You need to look at <http://scicomp.jlab.org/scicomp/> to check that your jobs are all done and that there were successful (and with EC=0).

---

-- [Calculate N\\_true and N\\_expected for each mass\\_bin](#)

**[19]** *click nTrue* (second column after clicking FITTING)

All jobs will be submitted to the farm. The time required depends on number of events and number of waves used. These jobs calculate the number of events expected to be observed and the true (nature produce) for each mass bin as predicted from the fit, for each wave and in total.

**WAIT** until everything is done in the farm.

You need to look at <http://scicomp.jlab.org/scicomp/> to check that your jobs are all done and that there were successful (and with EC=0).

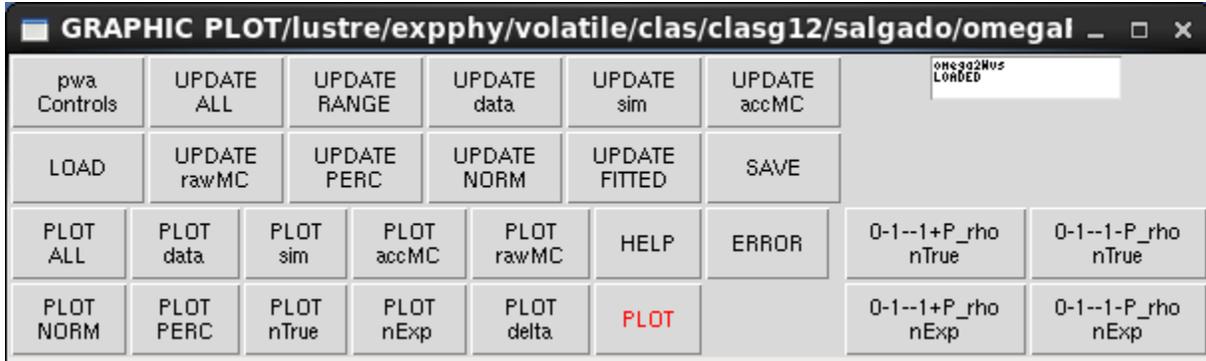
---

-- Produce plots (for total and for each wave)

From the main GUI go **back** and

**[20]** click **Graphic Plot**

This will open a GUI (takes a few seconds).



If it is the first time looking at your results you need to  
Click "UPDATE RANGE", "UPDATE data", "UPDATE accMC", "UPDATE rawMC", and  
"UPDATE FITTED(in that order) and then "SAVE"

Select what you want to plot (one or more distributions from the panel. All are plotted as function of mass bin.)

"PLOT" will plot all selected distributions. (see full description in documentation).

YOU ARE DONE WITH YOUR FIRST PWA FIT!

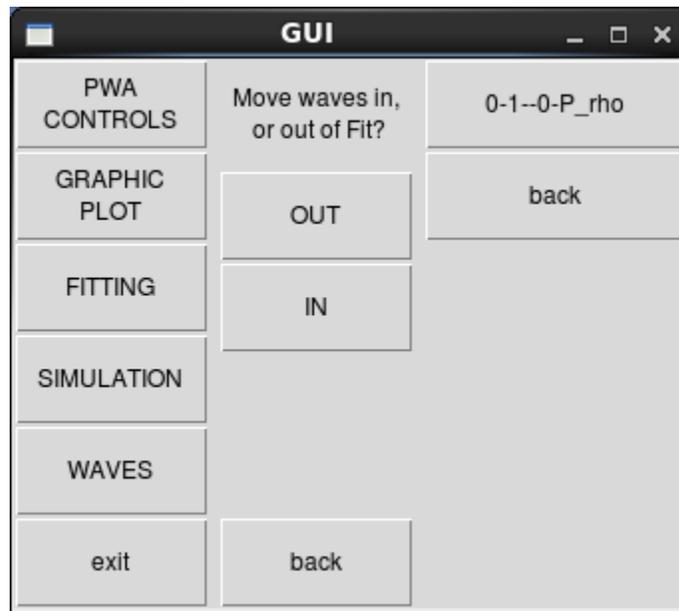
---

You are ready for your next PWA!! Changing the wave set in your fit.

---Changing wave set

From the main GUI

**[21]** *click* **WAVES**



A second column in the GUI will appear. It will ask you to move waves IN or OUT of your wave set in the fit; i.e. click OUT and a list of the waves you are currently using will pop up in a third column. Click on the waves you want to drop. Wait until the command line prints "DONE". For including new waves you need to have produced the *name.bamp* files, as specified before, and then click IN in similar way. You want then to *click* into **PWA CONTROLS** and change the name of the plotting file so that your old data will not be lost and you can compare different wave sets fittings later on.

After you include in the system the new wave set, you need to rerun **normint**, the **fitter**, and **nTrue**. Compare plots and interact as many times as needed.

-----