Weekly Report

Christina

HPC Monte Carlo + Poster & Proposal + Proton Event Classification

monte_carlo.py

1. Runs Monte Carlo algorithm on each event of a run.
2. Records each event's total chi^2 value on a .h5 file
3. Multiple try/except in order to make sure each event gets processed; the most common error comes from the MC Fitting step, because our cleaning process wipes out a significant amount of noisey events.

submit_jobs_105-119.sh

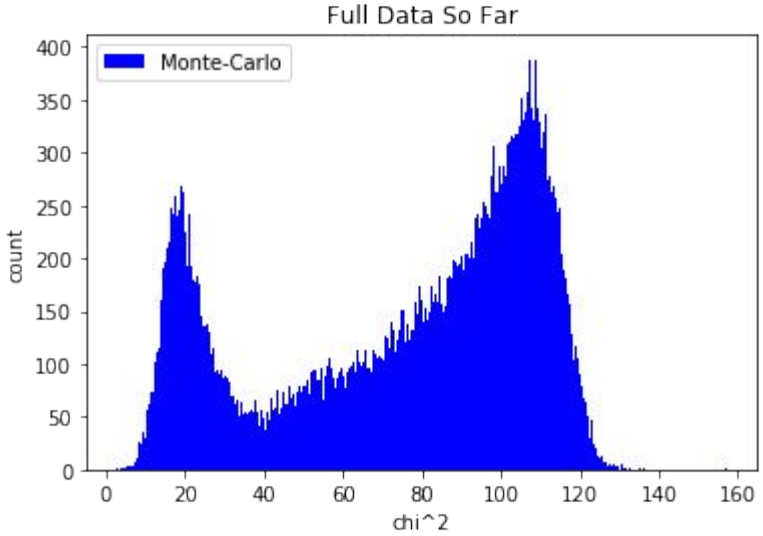
1. Submits 15 jobs (run 105 to run 119)
2. Uses a for loop
    a. Each iteration passes the run_number variable to the job script (monte_carlo 105-119.sh)
    b. Then submits (qsub) the job script before updating it with a new run number

monte_carlo_105-119.sh

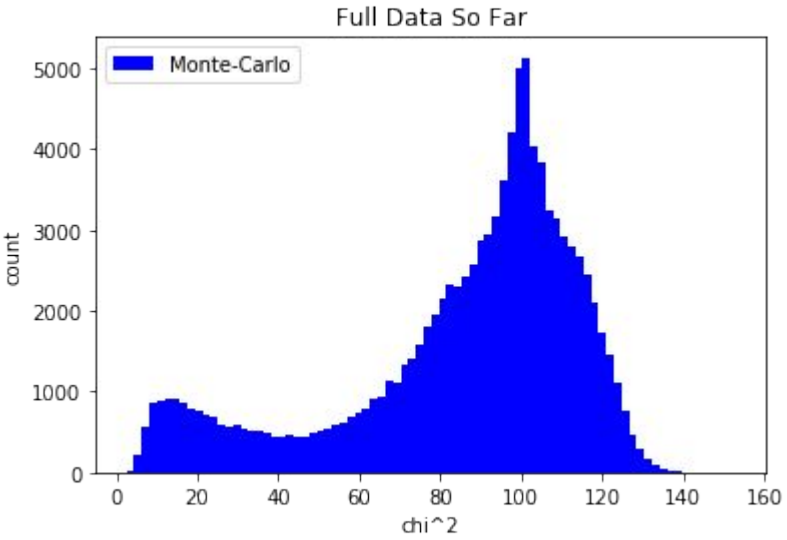
1. Where the PBS commands are
2. Runs the python file, and saves the .h5 file (of Monte Carlo results) to output directory

Results

40Ar


Full Data So Far
Monte-Carlo
count
chi^2
0
50
100
150
200
250
300
350
400
0
20
40
60
80
100
120
140
160

46Ar


Full Data So Far
Monte-Carlo
count
chi^2
0
1000
2000
3000
4000
5000
0
20
40
60
80
100
120
140
160

Reason for large chi^2 values:
Pile-up events
Shift in z direction due to trigger delay

Proton Classification

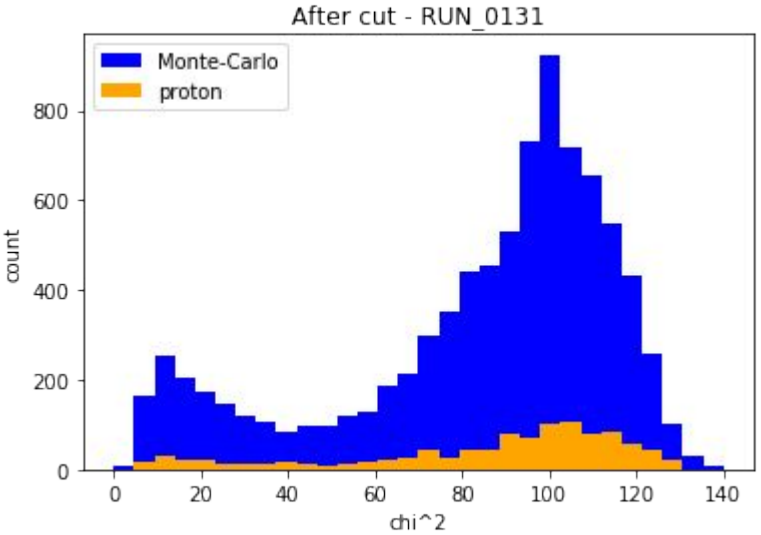
Python Files

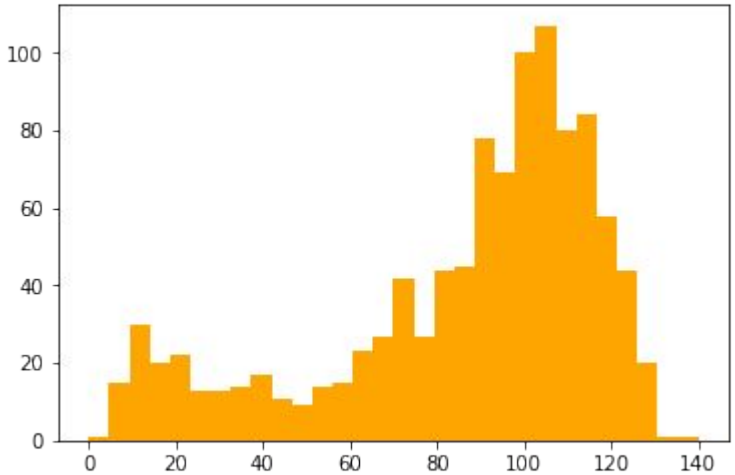
1. `bulkDisctetize_RealData.py`: discretizes the real training data, labeled from two runs of 46Ar.
2. `bulkDiscretize.py` discretizes all events in a .h5 file in 20*20*20 dimension
3. `basicNN_real2real_multic.py` divides the data to three event categories (proton, carbon and junk), while `NN_pCnoise.py` divides it to two categories (proton and non-proton).
4. `NN_load model.py` loads the model and weights, and predicts the outcome of the runs we are testing.

Neural Network

1. Proton / not proton: 0.86 accuracy
2. Multi-class: 0.79 accuracy
3. Potential problem: empty objects (events) in a run, so the index is shifted
    a. I added an index space where the empty objects supposed to occur to the indices of proton events
    b. For example, if the proton indices are (1,2,3,4,5) and the empty event is 3, then the indices become (1,2,4,5,6) so it matches the ones of our regular events.

Testing the proton filter





The proton classification did not seem to be working; there is still many events with large objective function values.

Problems could come from the Keras prediction process, event cleaning, shift in z or pile-up.