Generator Interface Usage

Ian Hunt-Isaak
ihuntisa@oberlin.edu (till May 2017)
ianhuntisaak@gmail.com (permanent)

This document provides complete instructions on how to set use the Monte Carlo Generators I set up this summer. They are availiable at `github.com/ianhi/GeneratorInterface/` Bash scripts set up to run these as batch jobs with varying pthat ranges are availiable at `github.com/ianhi/BatchJobs/`

1 Generator Interface

This set up allows for running PYTHIA, Jewel, Pyquen, Pyquen-Wide, and Q-Pythia through cmsRun and gives a root file with trees of jet information. It was originally set up by Yen-Jie Lee: `github.com/yenjie/HIGenerator` and modified by me. The basic setup is preserved in my verision but options are more easily accessible and Jets are reconstructed using both $R = .3$ and $R = .5$. My verision is availiable at `github.com/ianhi/GeneratorInterface`

1.1 Set up

To set up for usage:

```
$ cmsrel CMSSW_5_3_20/
$ cd CMSSW_5_3_20/src
$ cmsenv
$ git clone git@github.com/ianhi/GeneratorInterface.git
$ cd GeneratorInterface/
$ scram b -j20
```

scram b will compile the several c++ components in plugins available to python configuration files. Every time you modify a file that is not one of the python configuration files you will need to call scram b again to have your changes take effect.

1.2 Usage

To run a Generator navigate to one of the test sub directories, QPythiaInterface/test/ or JewelInterface/test/ and call cmsRun testGenName.py. I have added several optional command line arguments to allow choice of output file name, number of events, and pthat range. These options and their defaults are listed below.

Option	Default	cell3
maxEvents	2000	Maximum # of events
ptHatLow	120	Minimum P_T Hat
ptHatHigh	160	Maximum P_T Hat
output	[GenName]_DEFAULT_[Date and Time]_numEvent[# events].root	Default output name

Here is an example:

cmsRun testJewelDijet.py output="NAME" maxEvents=5000 ptHatLow=30 ptHatHigh=50

1.3 Modifications of Yen-Jie's set up

The major differences between this verision of Generator Interface and the version set up by Yen-Jie are the command line options and the trees produced.

The file: GeneratorInterface/JetNtupleproducer/src/DijetNtupleProducer.cc contains the code that generates the the trees and ntuples in the output root file. In Yen-Jie's version the two trees created are t and nt, where t contains the results of anti-K_T with $R = .3$. My updates results in the output file containing $t3$, $t5$, and nt. The number denotes the value of R used for anti-K_T, e.g. $t5$ corresponds to $R = .5$

2 Batch Jobs

Running the Generators in a batch job is necessary if you want to have a appreciable number of events and multiple $\hat{P_T}$ ranges. Bash scripts for this purpose are available at

`https://github.com/ianhi/BatchJobs`

These scripts will create a folder for each job in [..]/CMSSW_5_3_20/src/outputs. This was originally implemented due to the fact the Jewel Generator creates out.log and out.hepmc files and multiple jobs would compete for these files,

thus for Jewel this is set up is necessary to run multiple jobs and for the other generators this is done for consistency.

QPYTHIA MAYBE NEEDS THIS TO NOT HAPPEN.

2.1 Set Up - Usage

Perform the following commands :

```
$ mkdir outputs/ outputs/ROOT outputs/LOG
$ cd [PATH]/BatchJobs/
$ bash [GenName]_BatchScript.sh
```

Where [GenName]$\in$ {Pythia, Jewel, Pyquen, PyquenWide, QPythia}

3 $\hat{P_T}$ Weighting

When using combining simulations with different $\hat{P_T}$ bins you need to weight each bin according to a cross section.

3.1 Extracting Cross Section

The Cross sections for the various generators can be found in the output files.

3.2 Jewel

Jewel places this information in the out.log file

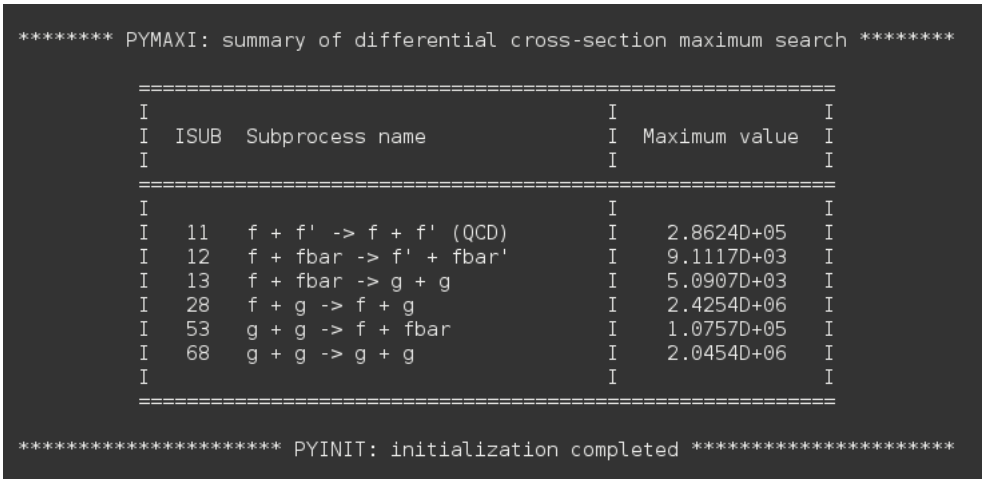

Figure 1: Cross section information in Jewel log file

The sum of all the cross sections will give you. You can extract and sum the cross section values for multiple out.log files using readOutLog.py. Cross section values for the bins I used are as follows:

Bracketed:
Bins:{15,30,50,80,120,170,220,280,330,400,460,540}
Weights: {4878812.4,4218696.2,4659866.4,4049616.0,2934218.1, 1695768.4,1302251.3,718081.6,578537.1,291444.26,168207.79}

3.3 Code

Loops to do the weighting can be found at: `https://github.com/ianhi/MonteMacros/blob/master/jtptWeight.C` Lines 104-127 and 198-208