GENIE: The Neutrino Event Generator

Amit Pal
NISER, Bhubaneswar
amit.pal@niser.ac.in
October 15, 2025

Outline

- Documentation and Installation
- Brief introduction of Event Generator and GENIE
- Cross-sections splines
- Event generation
- Tuning in GENIE
- Four exercise:
    - Cross-section plots
    - Kinematics distribution
    - Neutrino oscillation
    - Reconstruction of neutrino energy

GENIE Documentation

Genie **A Leading Neutrino Event Generator and Global Analysis of Scattering Data**

Predictive Simulations Supporting Precision and Discovery in Neutrino Physics

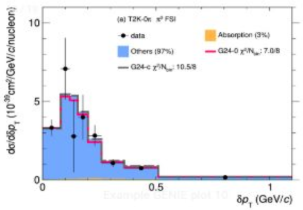
- Home
- Mission statement
- GENIE collaboration
- Policy documents
- GENIE products
- Copyright notices
- Citing GENIE
- Logos
- **Public releases**
- **Global fits & tunes**
- Naming conventions
- Data releases
- User forum
- Project incubator
- Physics & user manual
- Document database
- Publications & talks
- Slack workspace
- User mailing list
- Developer mailing list

GENIE is the world's leading neutrino event generator, **bridging theory and experiment in modern neutrino physics**. It underpins data interpretation and exploitation across major experiments.

The **international GENIE collaboration** plays the leading role in

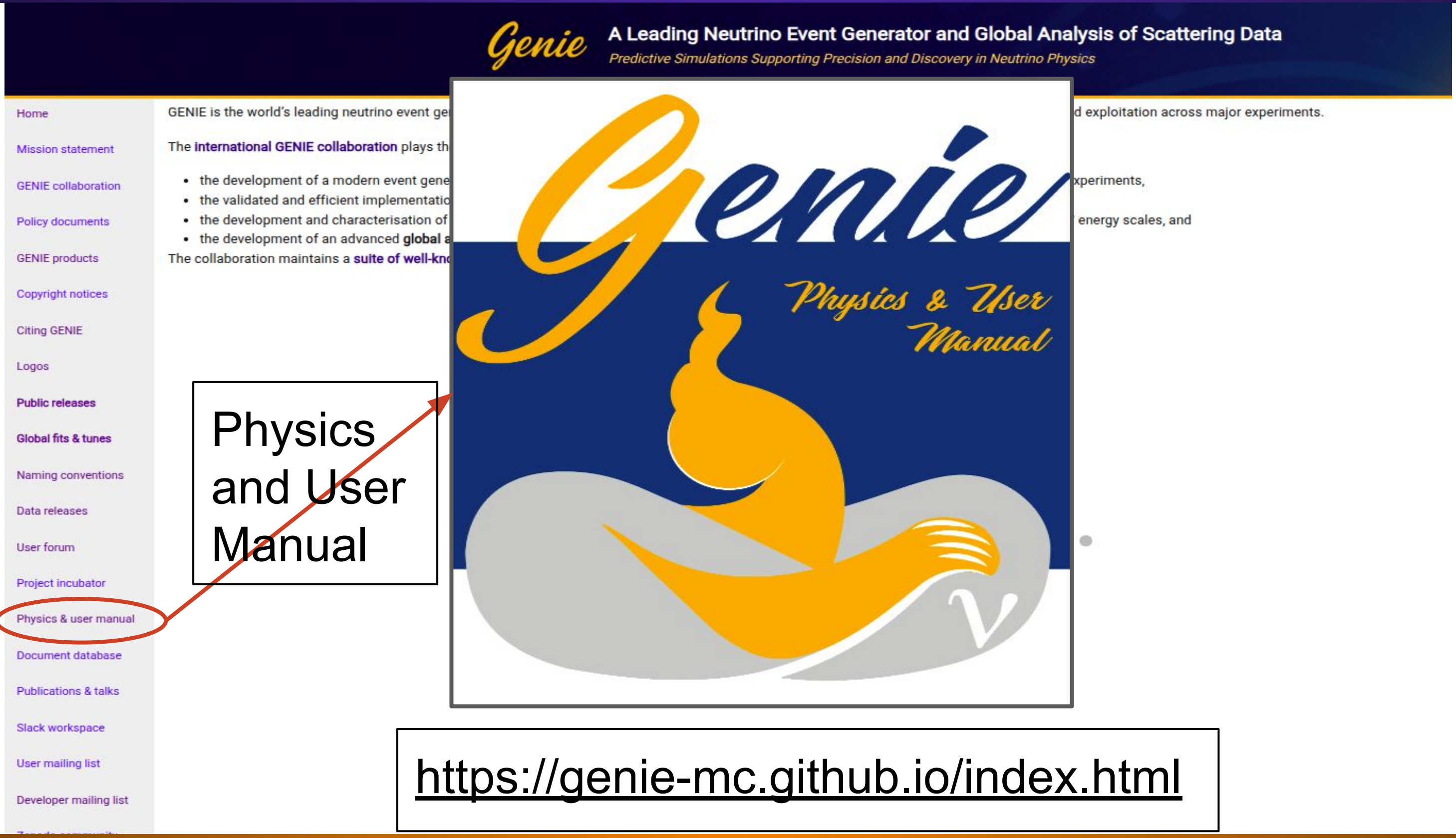
- the development of a modern event generation framework, including experimental interfaces and analysis-related tools in support of neutrino experiments,
- the validated and efficient implementation of a constellation of alternative physics models within a common platform,
- the development and characterisation of novel and predictive **comprehensive physics models** including all processes relevant from MeV to PeV energy scales, and
- the development of an advanced **global analysis of neutrino scattering data** for model tuning and data-driven model uncertainty evaluation.

The collaboration maintains a **suite of well-known software products**.



https://genie-mc.github.io/index.html

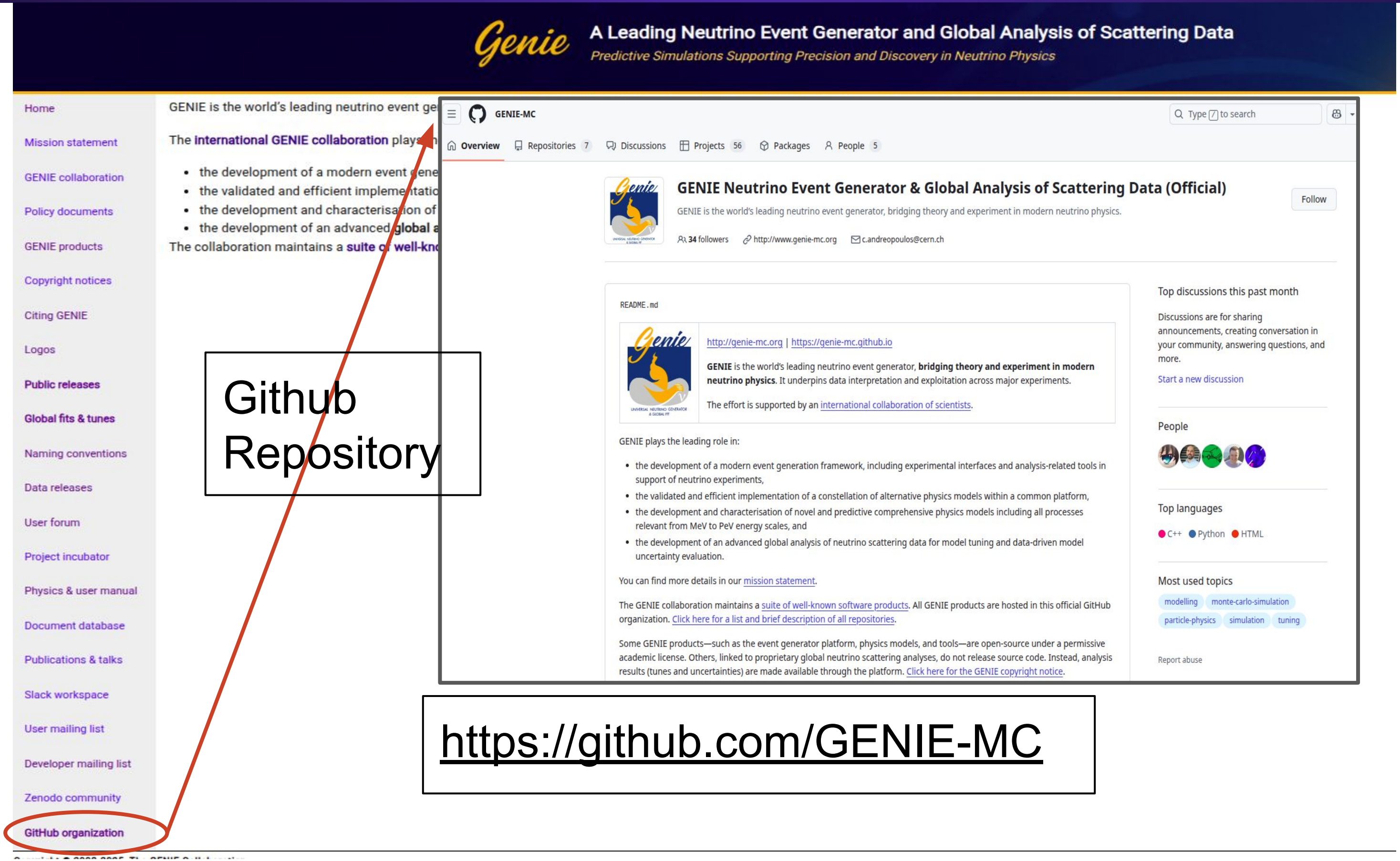
GENIE Documentation



Physics and User Manual

https://genie-mc.github.io/index.html

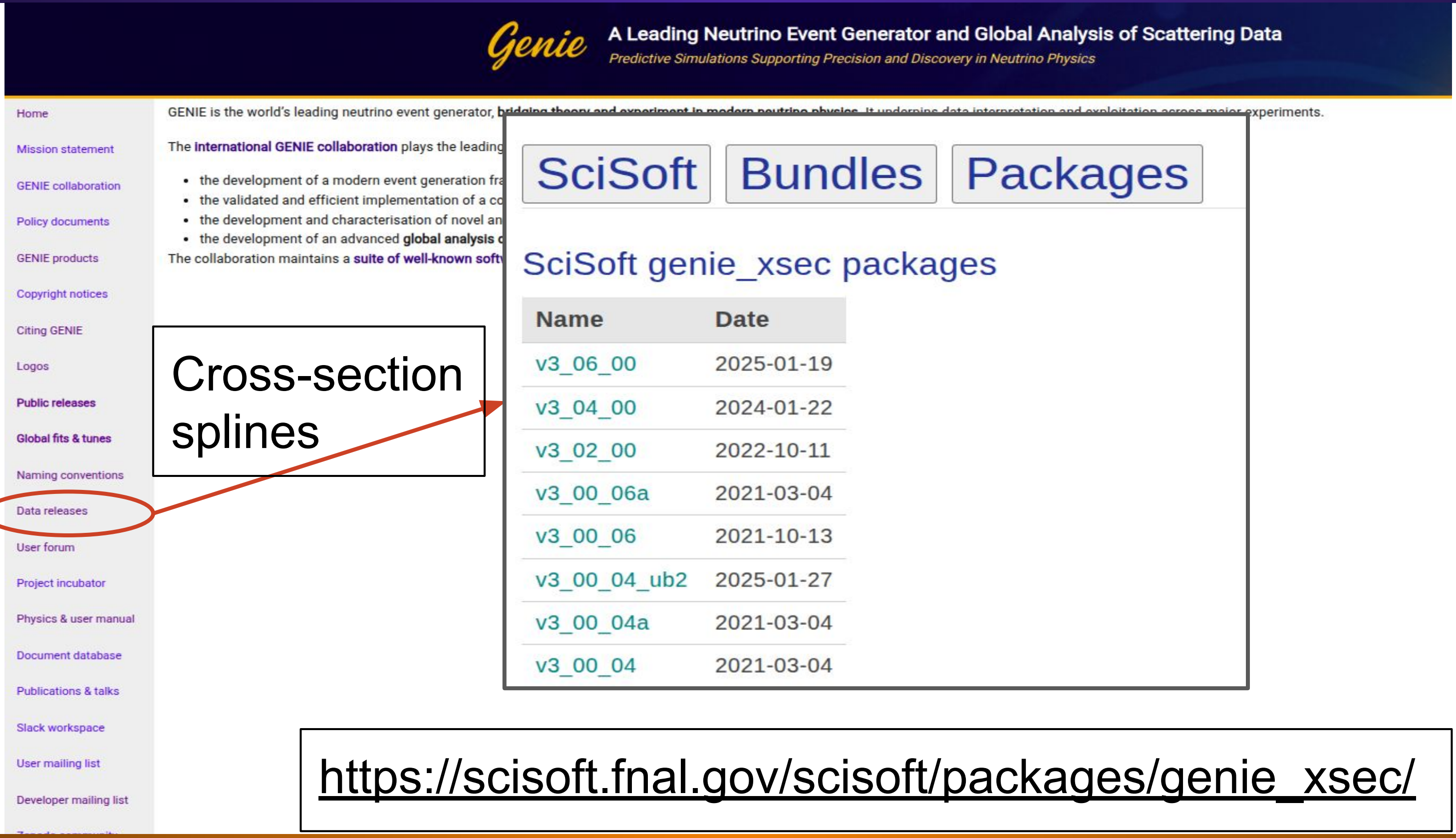
GENIE Documentation



Github Repository

https://github.com/GENIE-MC

GENIE Documentation



Cross-section splines

https://scisoft.fnal.gov/scisoft/packages/genie_xsec/

GENIE Installation

- The official GENIE source code is maintained at a Git repository: https://github.com/GENIE-MC/Generator
- To checkout the Generator:

```
$ git clone git@github.com:GENIE-MC/Generator.git
```

- This will clone the master branch. But you need to checkout a specific version.

```
$ git checkout R-3.04.00
```

- This gives you GENIE version *3.04.00*
- Then you need to setup environment variables for required libraries and then build it.

GENIE Installation

- GENIE installation require few external packages:
  - **GSL:** The GNU Scientific Library
    http://www.gnu.org/software/gsl/
  - **PYTHIA6:** The well-known Monte Carlo package
    https://pythia6.hepforge.org/
  - **ROOT:** A very well-known software for particle physics data analysis
    https://root.cern.ch/
    - ROOT should be built with GSL (MathMore) and PYTHIA6 support.
  - **LHAPDF5:** Standard tool for evaluating parton distribution functions (PDFs) in high-energy physics
    https://lhapdf.hepforge.org/
  - **log4cpp:** A C++ library for message logging.
  - **libxml2:** The C XML library for the GNOME project

GENIE Installation

- GENIE installation require few external packages:
  - **log4cpp:** A C++ library for message logging. http://log4cpp.sourceforge.net/
  - **libxml2:** The C XML library for the GNOME project http://www.xmlsoft.org/

GENIE Installation

- GENIE installation require few external packages:
  - **log4cpp:** A C++ library for message logging. http://log4cpp.sourceforge.net/
  - **libxml2:** The C XML library for the GNOME project http://www.xmlsoft.org/
- **But this is not very straightforward. You have to be very careful of the version of each of these packages.**
- **Very often, you might encounter conflict of various libraries.**
- An easy solution: using **Apptainer/Container**
- I build everything within an independent **Apptainer** environment and created a **sif** image (*genie.sif*).
  - It has **root, genie**,.. etc installed within it
- It is available in Google Drive, you need to download it and open with apptainer.
- You are ready to go 🤗

GENIE Installation

- But you need to have installed Apptainer in your system.
- Apptainer installation:
  - **Mac:** Apptainer is available via *Lima*. It can be installed using *brew*. Follow instructions from https://apptainer.org/docs/admin/1.4/installation.html#mac
  - **Windows:** It can be installed with WSL2. To install geant4, probably some of you already installed WSL2. Use that one. Then please follow: https://apptainer.org/docs/admin/1.4/installation.html#windows
  - **Linux system** -
    - **Ubuntu:**

```
$ sudo apt update
$ sudo apt install -y software-properties-common
$ sudo add-apt-repository -y ppa:apptainer/ppa
$ sudo apt update
$ sudo apt install -y apptainer
```

GENIE Installation

- But you need to have installed Apptainer in your system.
- Apptainer installation:
  - **Linux system -**
    - **Fedora:**

```
$ sudo dnf install -y apptainer
```

    - **AlmaLinux/CentOS:**

```
$ sudo dnf install -y epel-release
$ sudo dnf install -y apptainer
```

- You can check apptainer installation:

```
$ apptainer --version
```

GENIE Installation

- Assuming you have already downloaded the ***genie.sif*** file, make a sandbox out of it

```
$ apptainer build --sandbox genie_sandbox/ genie.sif
```

- Enter the apptainer shell

```
$ apptainer shell --writable genie_sandbox/
```

- Set environment for GENIE

```
$ cd /opt/mywork/
$ source do_end_genie.sh
```

- Generate first event using GENIE (just for test)

```
$ cd /opt/mywork/
$ gevgen -r 3 -n 100 -p 14 -t 1000010020 -e 1.0
--cross-sections gxspl-NUsmall.xml
```

Clone Git Repository

If you haven’t cloned before, use:

```
$ git clone
https://github.com/amitpal96/MC_tutorial.git
```

If you cloned before during the installation:

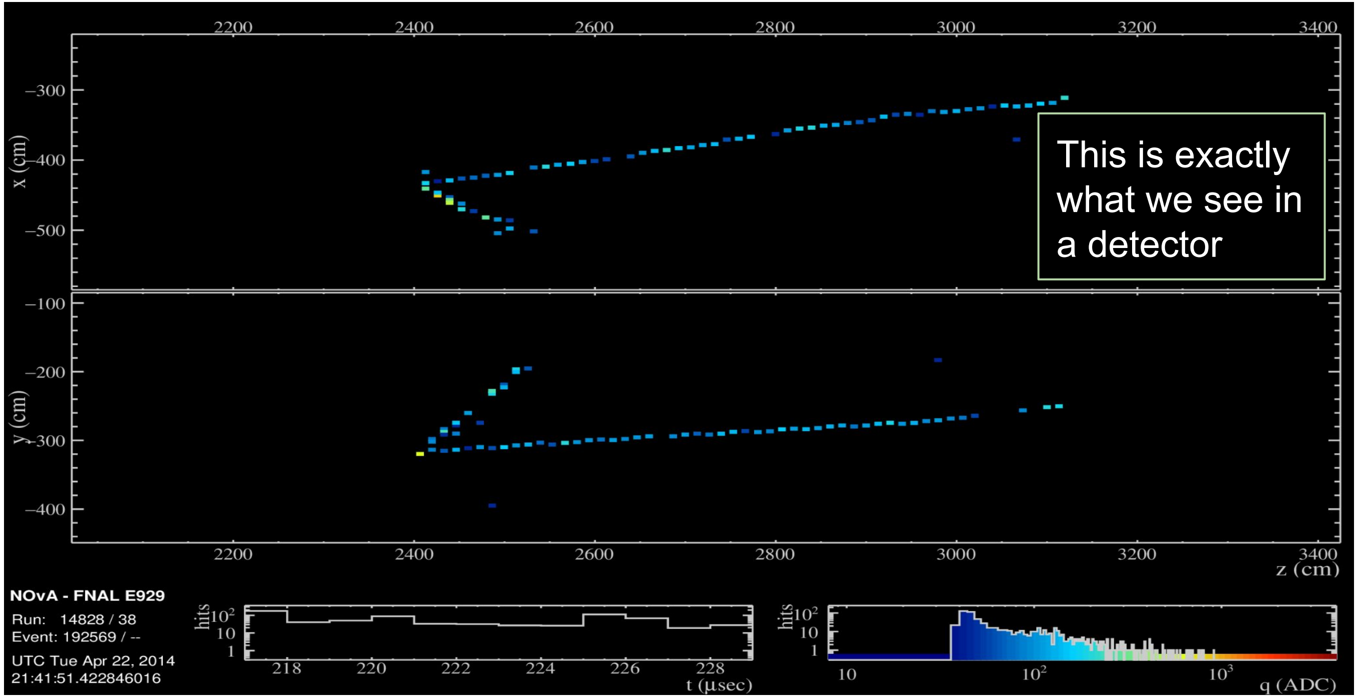
```
$ git pull origin main
```

Why do we need Event Generators?

- Neutrino interact **rarely** and **weakly** – direct measurement of interaction kinematics is hard.
- Beam is not **monoenergetic** – must infer neutrino energy from observed particles.
- Detectors measure final states, not **interaction types** directly.

Why do we need Event Generators?

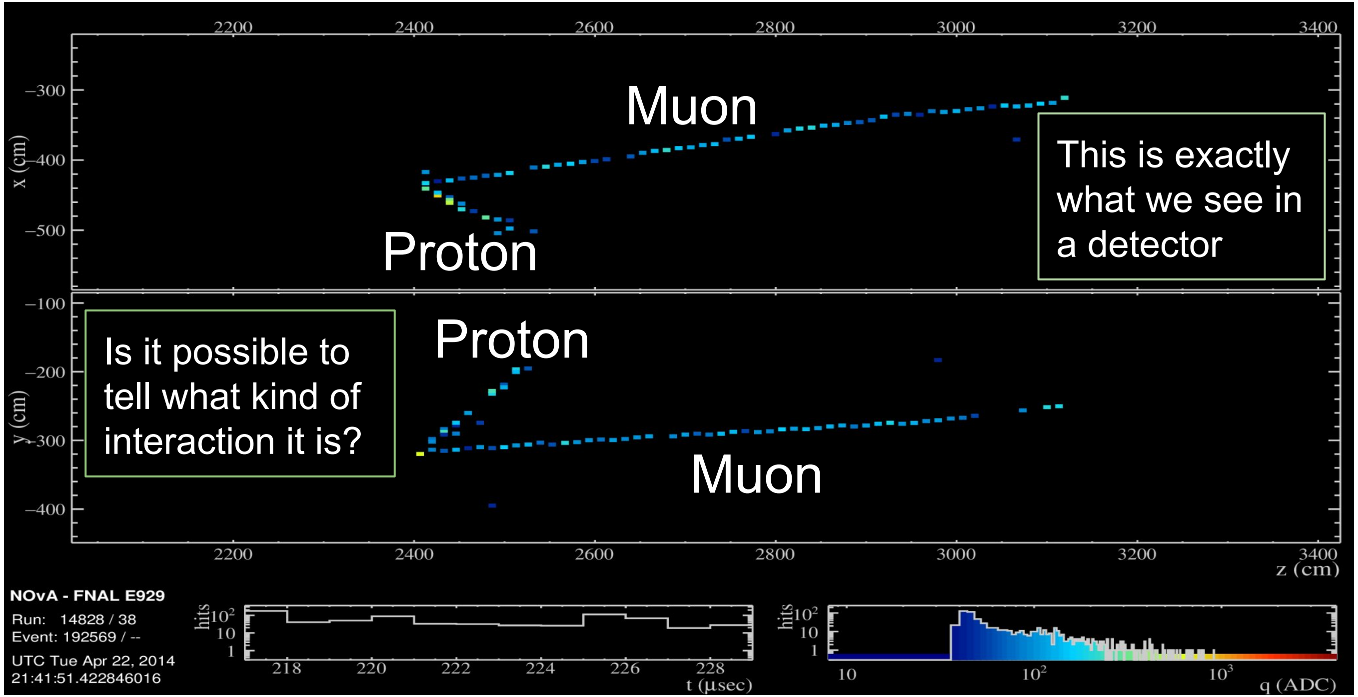
- Neutrino interact **rarely** and **weakly** – direct measurement of interaction kinematics is hard.
- Beam is not **monoenergetic** – must infer neutrino energy from observed particles.
- Detectors measure final states, not **interaction types** directly.



NOvA Event Display

Why do we need Event Generators?

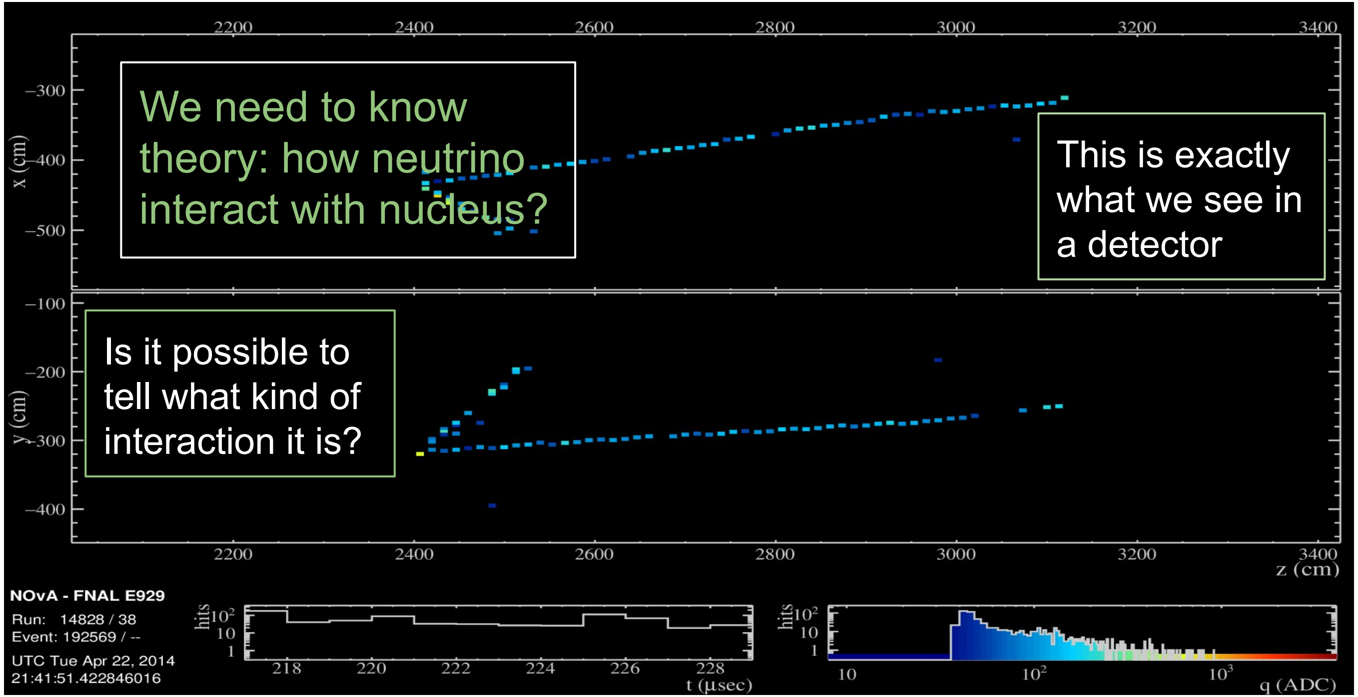
- Neutrino interact **rarely** and **weakly** – direct measurement of interaction kinematics is hard.
- Beam is not **monoenergetic** – must infer neutrino energy from observed particles.
- Detectors measure final states, not **interaction types** directly.



NOvA Event Display

Why do we need Event Generators?

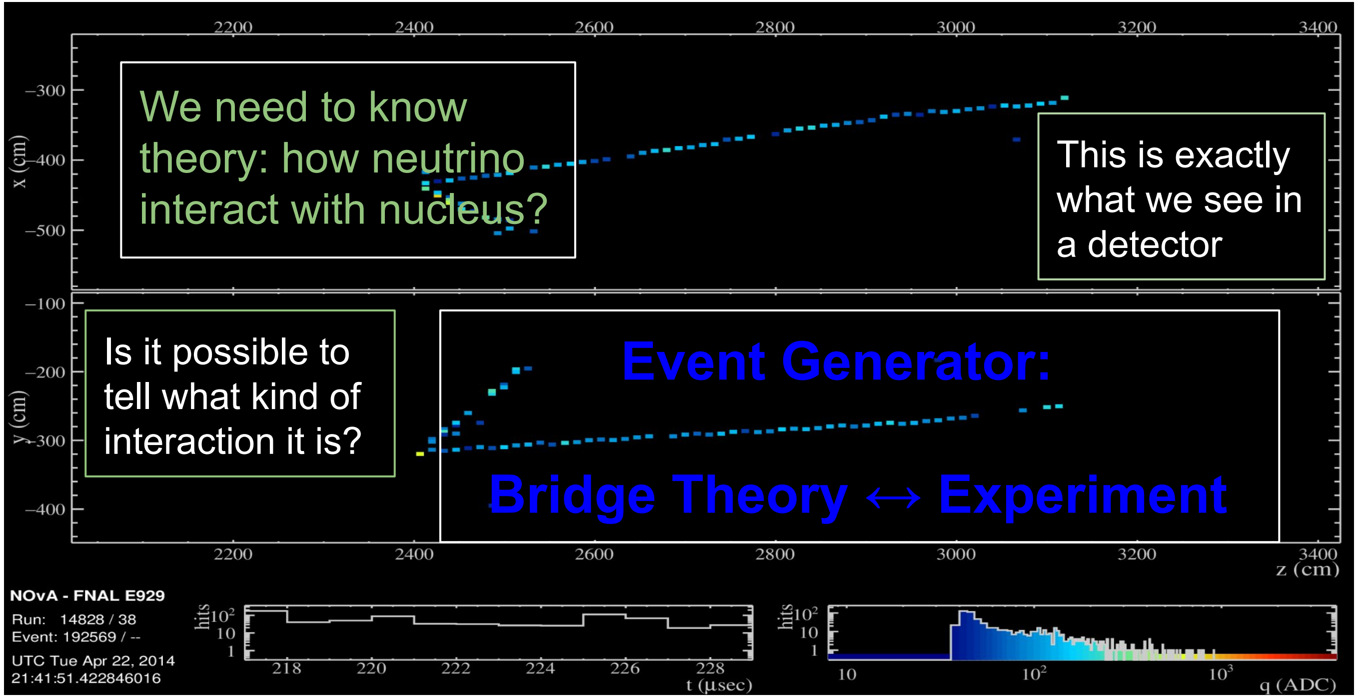
- Neutrino interact **rarely** and **weakly** – direct measurement of interaction kinematics is hard.
- Beam is not **monoenergetic** – must infer neutrino energy from observed particles.
- Detectors measure final states, not **interaction types** directly.



NOvA Event Display

Why do we need Event Generators?

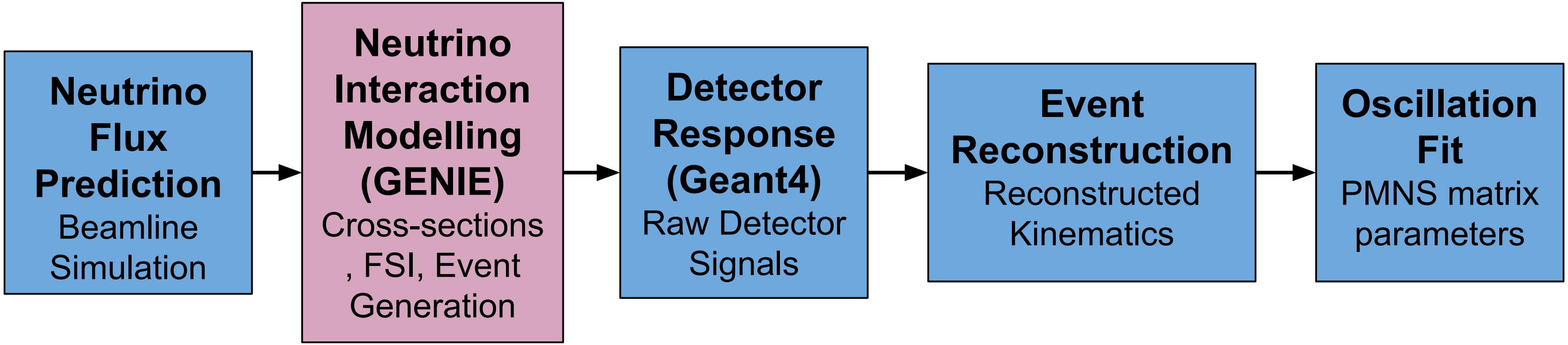
- Neutrino interact **rarely** and **weakly** – direct measurement of interaction kinematics is hard.
- Beam is not **monoenergetic** – must infer neutrino energy from observed particles.
- Detectors measure final states, not **interaction types** directly.



NOvA Event Display

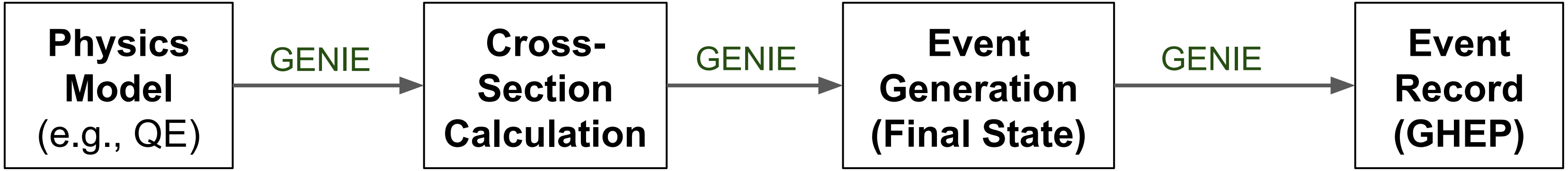
Role of Event Generators in Experiments

- **Neutrino Flux Prediction:** Determines the incoming ν energy and flavor at the detector.
- **Interaction Modeling (GENIE):** Simulates the ν-nucleus scattering, dictating the type and kinematics of the final-state particles.
  - This is the physics input!
- **Detector Response Simulation (e.g., Geant4):** Tracks the generated particles and models the detector's observable signals
- **Event Reconstruction:** Uses detector signals to infer particle properties and the original interaction (e.g., muon momentum, vertex location).
- **Physics Analysis:** Applies the reconstructed data to extract parameters like $sin^2\theta_{23}$ or measure cross-sections.



What Event Generators Do

- **Simulate neutrino–nucleus interactions:**
  - **Select Interaction Channel:** Determines the physics process based on probability (e.g., CCQE, RES, DIS, COH, 2p2h).
  - **Calculate Kinematics:** Determines the momentum, energy, and angle for all outgoing particles (lepton, hadrons).
  - **Simulate Intranuclear Transport (FSI):** Models how primary and secondary hadrons (protons, pions, etc.) re-scatter, get absorbed, or exchange charge while exiting the nucleus.
- **Generate Event Record:** Produces the complete history of the interaction in a standard format (GHEP, which is then often converted to ROOT/TTree).
- **Enable Systematic Studies:** Allows variation of physics parameters (tunes) or switching models to evaluate theoretical uncertainties.



Available Neutrino Event Generators

Genie
UNIVERSAL NEUTRINO GENERATOR & GLOBAL FIT

Used by Fermilab Experiments (MINOS, NOvA, MINERvA, DUNE)

NEUT

Used by T2K, SK and H2K

ACHILLES

Very new, theory driven, led by Fermilab

GiBUU

Theory driven, most sophisticated FSI model

NuWro * Wrocław Neutrino Event Generator *

Theory driven with many model options, early adopter of new theory developments

What is GENIE?

- A **Neutrino Monte Carlo Generator**: written in C++.
- Simulates almost **all kinds of neutrino** interactions.
- Handles **all kind of neutrinos and target** materials.
- Compatible for **very low energy (MeV) to very high energy (PeV)** neutrino experiments.
- Provides options to input **real detector geometry.**
- Many tools for studying **systematics, comparison to data**.
- Provides direct access to **Model Parameters** to modify and test.
- Reweighting framework: Propagating **uncertainties/Systematic** errors.

GENIE Event Generation Steps

Compute neutrino-nucleus cross-sections

↓

Prepare/decide input parameters for event generation

↓

Generate neutrino events

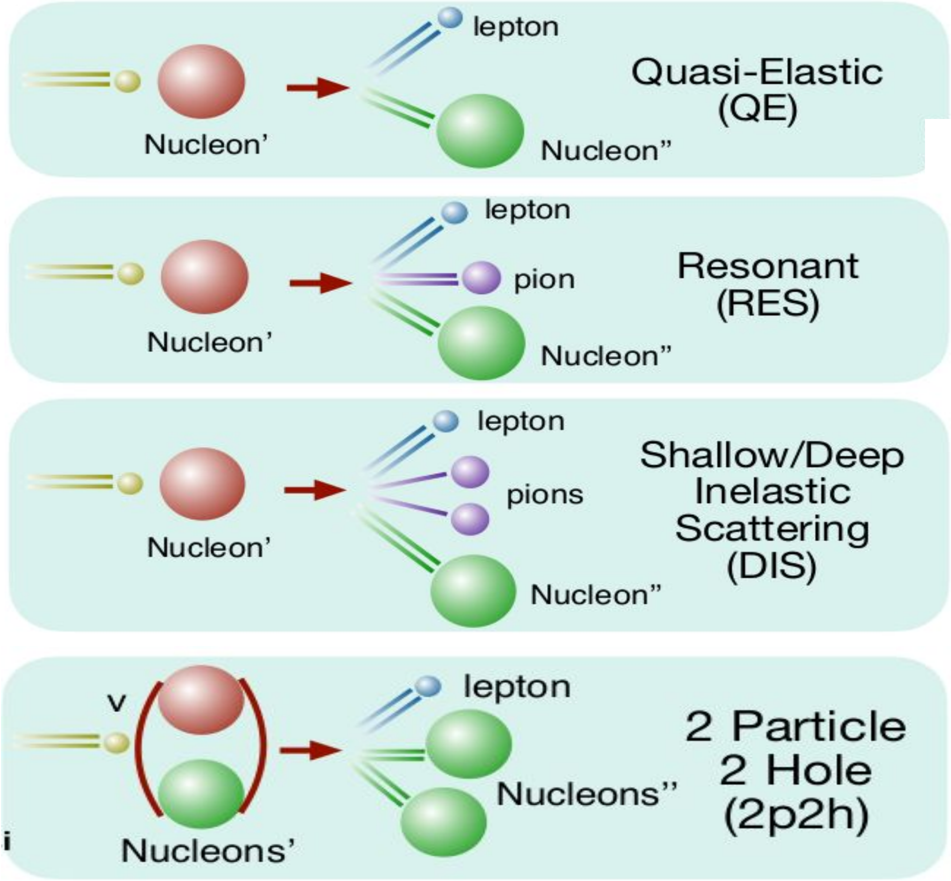
↓

Convert genie output into a root file

↓

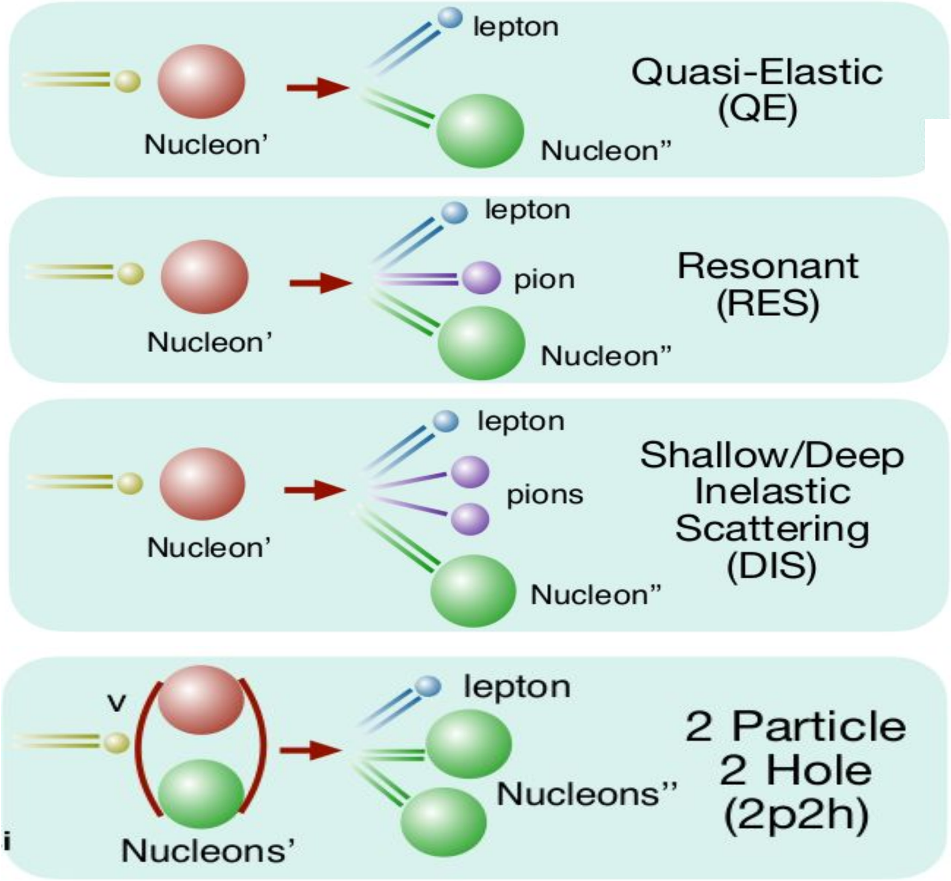
Analyze root files for Physics results

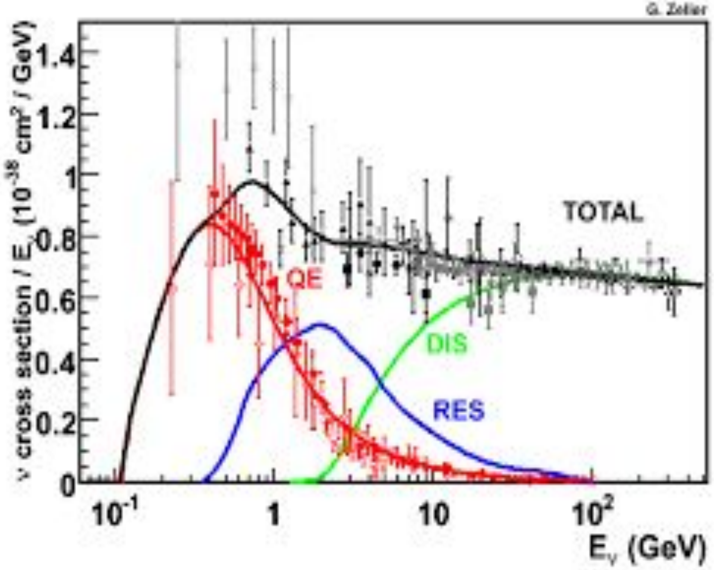
Neutrino-Nucleus interaction cross-section



Graphics by L. Cremonesi

Neutrino-Nucleus interaction cross-section

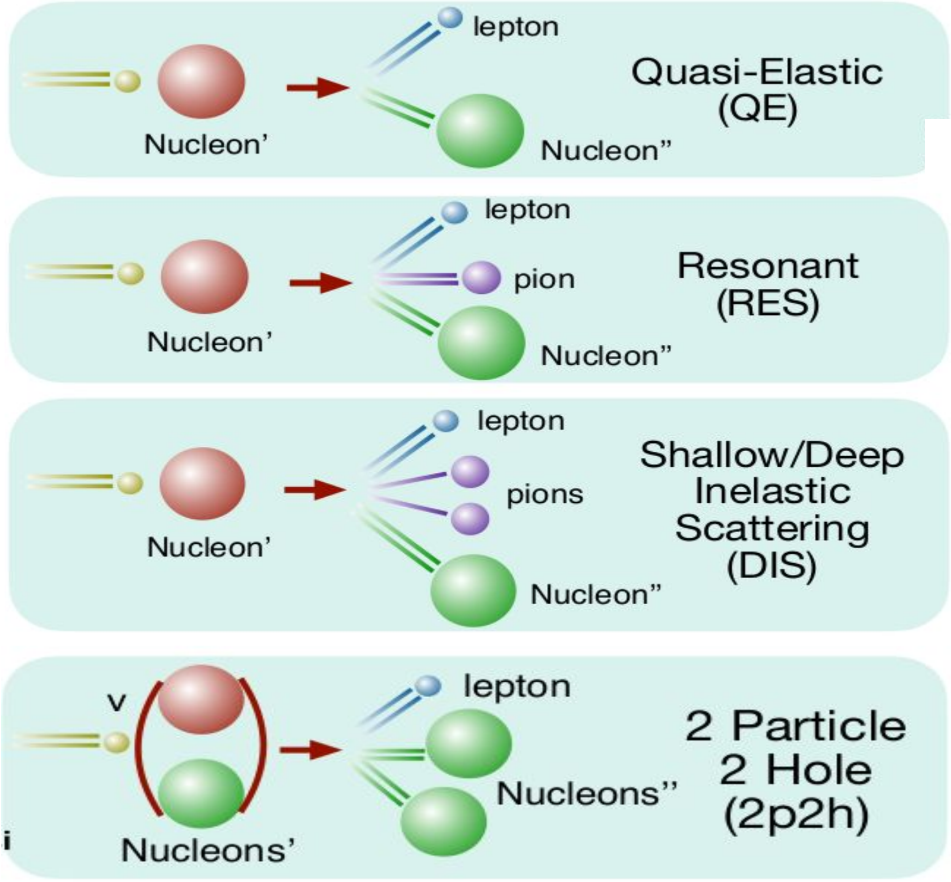




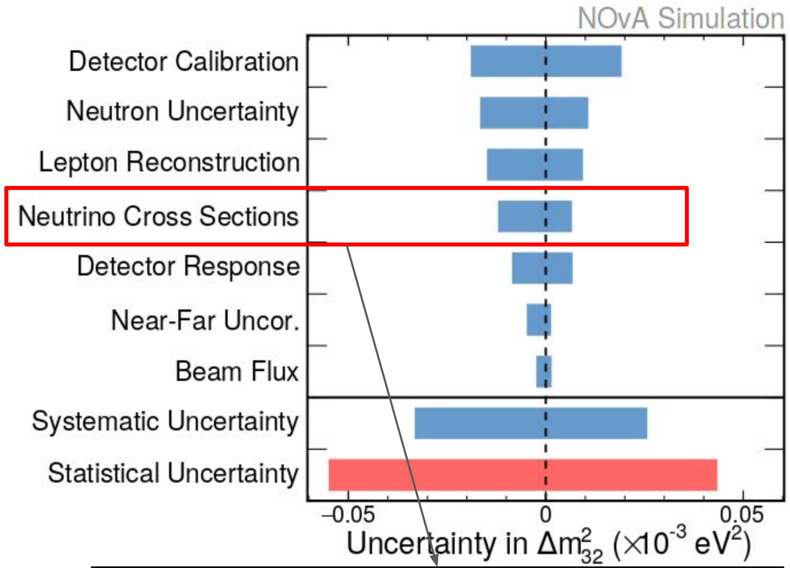
$$\sigma^{tot} = \sigma^{QEL} \oplus \sigma^{1\pi} \oplus \sigma^{2\pi} \oplus \ldots \oplus \sigma^{1K} \oplus \ldots \oplus \sigma^{DIS}$$

Graphics by L. Cremonesi

Neutrino-Nucleus interaction cross-section



Graphics by L. Cremonesi



One of the largest uncertainties in current long-baseline neutrino experiments

- Lots of models
- Different models at different energy range

Computing cross-section splines

- Neutrino interaction cross-section calculation is one of the **CPU-intensive process**.
- For a given process, **~10^9 differential cross-section calculation** is required.
- GENIE first computes cross-section separately and store in a xml file (called **Splines**)
- During event generation, it can **load the pre-computed cross-section files** and estimate the cross section by numerical interpolation
- There are a lots of pre-computed splines available.
- Highly encouraged to **download pre-computed splines from** https://scisoft.fnal.gov/scisoft/packages/genie_xsec/
- Make sure you are downloading for correct GENIE **version and required tune.**
- If your desired tune splines is not available in the webpage then only you should go for generating splines by yourself. That also should be once for all.

Step 1: Generating cross-section splines

```
$ gmkspl -p neutrino_code <-t target_codes, -f geometry> [-n nknots]
         [-e max_energy] [<--output-cross-sections | -o> xml_file]
         [--input-cross-sections xml_file] [--seed rnd_seed_num]
         [--event-generator-list list_name] [--message-thresholds xml_file ]
```

neutrino_code (PDG):	12 (-12), 14 (-14), 16 (-16)
target_codes (PDG):	100ZZZAAAI; eg- for **C12, 1000060120**
nknots:	Number of knots per spline.
max_energy:	Maximum neutrino energy in GeV
xml_file :	Name of the output xsec spline file
rnd_seed _num:	Random seed

```
$ gmkspl -p 14 -t 1000060120 -n 50 -e 100 –seed 171872 -o
xsec_spline_p14_t1000060120_e100.xml --tune G18_01a_00_000
```

Don’t run this command now!!!

We will use pre-computed splines.

Step 2: Extract cross-section from splines

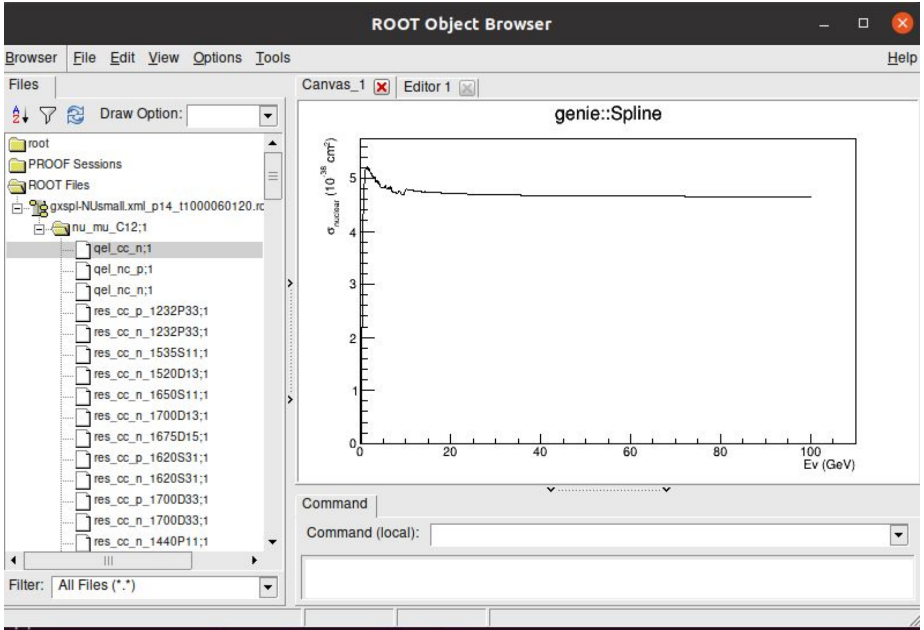
$ gspl2root -f input_xml_file -p neutrino_pdg_code -t target_pdg_code [-e maximum_energy] [-o output_root_file] [-w]

```
$ gspl2root -f
/opt/mywork/gxspl-NUsmall.xml -p
12,-12,14,-14,16,-16 -t 1000060120 -o
gxspl-NUsmall.xml_p14_t1000060120.roo
t --tune G18_02a_00_000
```

Step 2: Extract cross-section from splines

$ gspl2root -f input_xml_file -p neutrino_pdg_code -t target_pdg_code [-e maximum_energy] [-o output_root_file] [-w]

```
$ gspl2root -f /opt/mywork/gxspl-NUsmall.xml -p 12,-12,14,-14,16,-16 -t 1000060120 -o gxspl-NUsmall.xml_p14_t1000060120.root --tune G18_02a_00_000
```

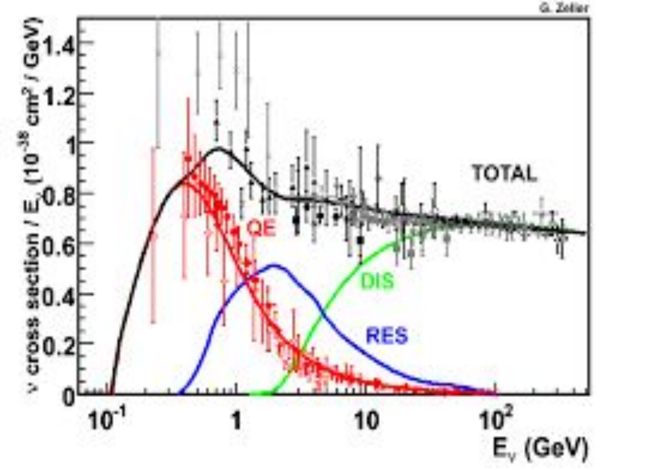


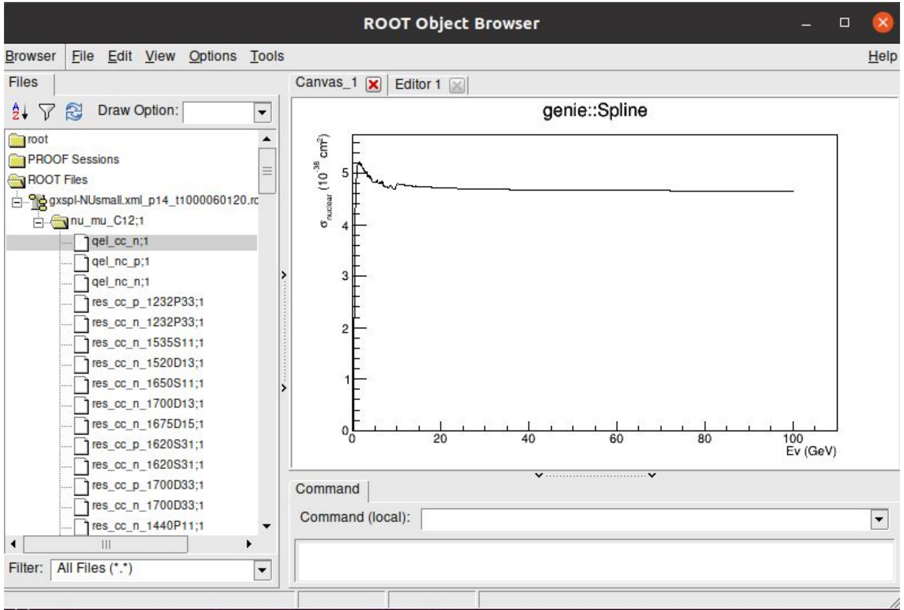

```
$ root -l gxspl-NUsmall.xml_p14_t1000060120.root
root[1] TBrowser t
```

Step 2: Extract cross-section from splines

$ gspl2root -f input_xml_file -p neutrino_pdg_code
-t targe ... o
output_

```
$ gsp ...
/opt/ ... -p
14 -t ...
gxspl ... 0120.
root --tune G18_02a_00_000
```

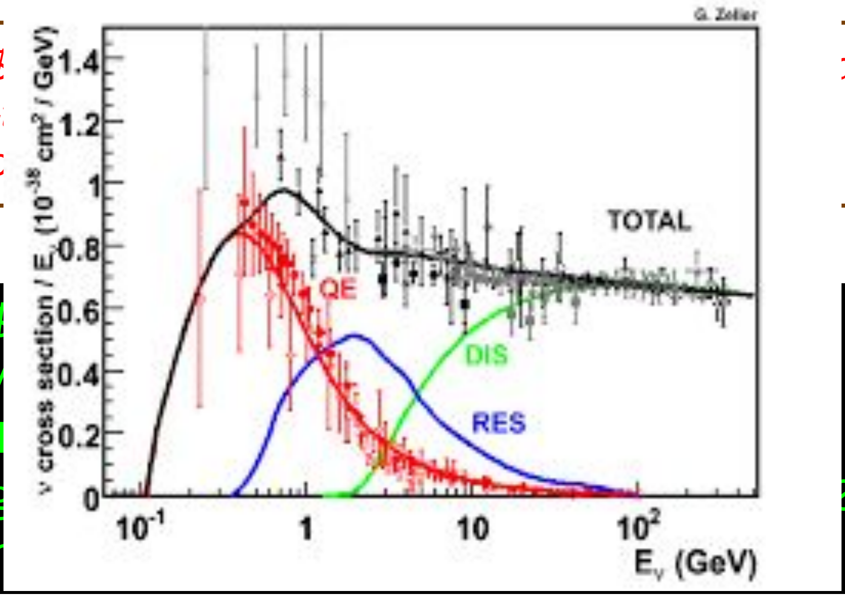





```
Apptainer> root -l gxspl-NUsmall.xml_p14_t1000060120.root
root [0]
Attaching file gxspl-NUsmall.xml_p14_t1000060120.root as _file0...
(TFile *) 0x563bfd5f99f0
root [1] TBrowser t
(TBrowser &) Name: Browser Title: ROOT Object Browser
root [2]
```

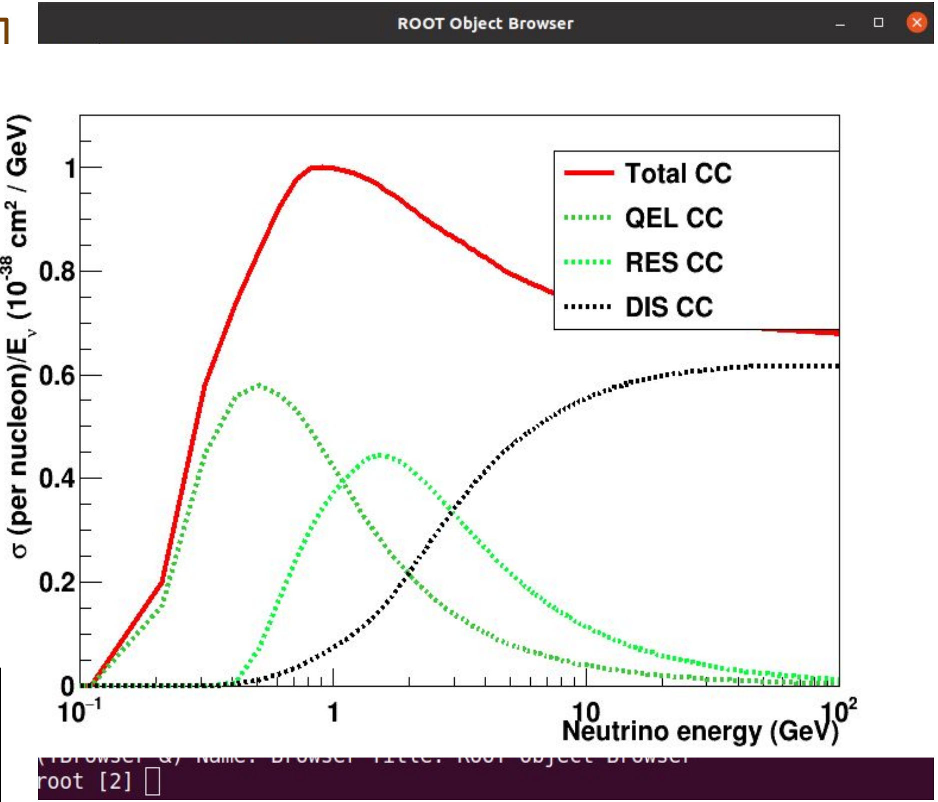
QE plot does not look similar?

Step 2: Extract cross-section from splines



Now look similar, right?

```
$ cd /opt/mywork/proj1/
$ root -l extract_xsec.cc
```



Then follow terminal prompt. This plot will appear in plots/ directory.

Step 3: Generate First Neutrino Event

$ gevgen [-h] [-r run#] -n nev -p neutrino_pdg -t target_pdg -e energy [-f flux]
[-w] [-seed random_number_seed] [--cross-sections xml_file] [--tune EX00_00a_00_000] [--event-generator-list list_name]
[--message-thresholds xml_file] [--unphysical-event-mask mask]
[--event-record-print-level level] [--mc-job-status-refresh-rate rate]
[--cache-file root_file]

[ ] are optional argument

run# : run number
nev : number of evets
neutrino_pdg : neutrino pdg (eg. 12, 14)
target_pdg : target pdg (100ZZZAAAI)
energy : energy in GeV. For a fixed energy provide one value, but for range use comma separated value (e.g: -e 0.5,5.0)
flux: neutrino flux spectrum

- as a function: -f 'sin(x)'
- as a vector file: -f fluxvec.data ; data file should contain two column corresponding to energy(GeV) and flux
- as a 1d histogram root file: -f flux.root,numu ; numu object in the root file

Step 3: Generate First Neutrino Event

```
$ gevgen [-h] [-r run#] -n nev -p neutrino_pdg -t target_pdg -e energy [-f flux]
         [-w] [-seed random_number_seed] [--cross-sections xml_file] [--tune
         EX00_00a_00_000] [--event-generator-list list_name]
         [--message-thresholds xml_file] [--unphysical-event-mask mask]
         [--event-record-print-level level] [--mc-job-status-refresh-rate rate]
         [--cache-file root_file]
```

[ ] are optional argument

random_number_seed : Random number
xml_file : Pre-computed xml splines
EX00_00a_00_000 : Cross-section tuning , eg: G18_02a_00_000
list_name : Specifies the list of event generators to use in the MC job
mc-job-status-refresh-rate : Allows users to customize the refresh rate of the status file

Tune in GENIE: Cross-section Models

Interaction Tune	Ground State	Quasi-elastic	2p2h	Resonance	DIS
G18_01{a,b,c,d}	RFG w/ NN tail	LS w/ dipole $F_A(Q^2)$	Dytman	RS tuned (2020)	BY tuned (2020)
G18_02{a,b,c,d}	RFG w/ NN tail	LS w/ dipole $F_A(Q^2)$	Dytman	BS tuned (2020)	BY tuned (2020)
G18_10{a,b,c,d}	LFG	NAV w/ dipole $F_A(Q^2)$	NSV	BS tuned (2020)	BY tuned (2020)
G18_10{i,j,k,l}	LFG	NAV w/ dipole $F_A(Q^2)$	NSV	BS tuned (2020)	BY tuned (2020)
G21_11{a,b,c,d}	LFG	SuSAv2 w/ dipole $F_A(Q^2)$	SuSAv2	BS tuned (2020)	BY tuned (2020)

- **Hadronization model:**
  - **AGKY:** C.Andreopoulos, H.Gallagher, P.Kehayias and T.Yang neutrino-induced hadronization model
- **FSI:**
  - **hA:** Effective intranuclear transport model in INTRANUKE
  - **hN:** Full intranuclear cascade (INC) model in INTRANUKE

- **RFG:** Relativistic Fermi Gas model
- **LFG:** Local Fermi Gas model
- **LS** - LLewellyn-Smith quasi-elastic model
- **NAV** - J. Nieves, J. Enrique Amaro, and M. Valverde quasi-elastic interaction model
- **NSV** - J. Nieves, I. Ruiz Simo, and M.J. Vicente Vacas 2p2h model
- **RS** - Rein and Seghal model for resonance
- **BS** - Berger and Seghal model for resonance
- **BY** - Bodek and Yang model

Tune in GENIE

- For each tune, there is a *$GENIE/config/G18_01a/ModelConfiguration.xml*
- All the physics models used for this tune are described in this xml file.

```
<param type="alg" name="XSecModel@genie::EventGenerator/QEL-CC">        genie::LwlynSmithQELCCPXSec/Dipole   </param>
<param type="alg" name="XSecModel@genie::EventGenerator/QEL-NC">        genie::AhrensNCELPXSec/Default        </param>
<param type="alg" name="XSecModel@genie::EventGenerator/QEL-EM">        genie::RosenbluthPXSec/Default        </param>
```

```
<param type="alg" name="XSecModel@genie::EventGenerator/RES-CC">        genie::ReinSehgalRESPXSec/NoPauliBlock      </param>
<param type="alg" name="XSecModel@genie::EventGenerator/RES-NC">        genie::ReinSehgalRESPXSec/NoPauliBlock      </param>
<param type="alg" name="XSecModel@genie::EventGenerator/RES-EM">        genie::ReinSehgalRESPXSec/EM-NoPauliBlock      </param>

<param type="alg" name="XSecModel@genie::EventGenerator/DIS-CC">        genie::KNOTunedQPMDISPXSec/Default          </param>
<param type="alg" name="XSecModel@genie::EventGenerator/DIS-NC">        genie::KNOTunedQPMDISPXSec/Default          </param>
<param type="alg" name="XSecModel@genie::EventGenerator/DIS-EM">        genie::KNOTunedQPMDISPXSec/Default          </param>
```

```
<param type="alg" name="XSecModel@genie::EventGenerator/MEC-NC">        genie::EmpiricalMECPXSec2015/Default  </param>
<param type="alg" name="XSecModel@genie::EventGenerator/MEC-EM">        genie::EmpiricalMECPXSec2015/Default  </param>
<param type="alg" name="XSecModel@genie::EventGenerator/NucleonDecay">  genie::DummyPXSec/Default             </param>
<param type="alg" name="XSecModel@genie::EventGenerator/NNBarOsc">      genie::NNBarOscDummyPXSec/Default     </param>
```

Tune in GENIE

- For each tune, there is a *$GENIE/config/G18_01a/ModelConfiguration.xml*
- All the physics models used for this tune are described in this xml file.

Lets focus on QEL-CC process only

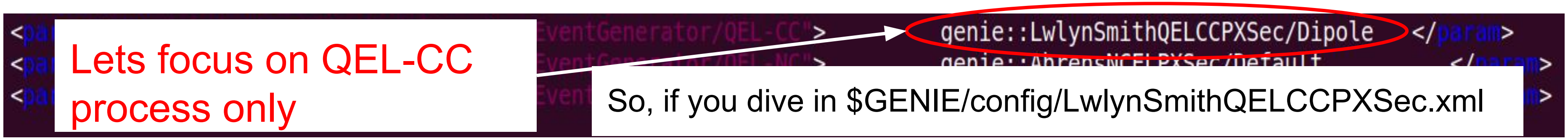
```
<pa  EventGenerator/QEL-CC">     genie::LlwynSmithQELCCPXSec/Dipole </param>
<pa  EventGenerator/QEL-NC">     genie::AhrensNCELPXSec/Default       </param>
<pa  EventGenerator/QEL-EM">     genie::RosenbluthPXSec/Default       </param>
```

```
<param type="alg" name="XSecModel@genie::EventGenerator/RES-CC">     genie::ReinSehgalRESPXSec/NoPauliBlock     </param>
<param type="alg" name="XSecModel@genie::EventGenerator/RES-NC">     genie::ReinSehgalRESPXSec/NoPauliBlock     </param>
<param type="alg" name="XSecModel@genie::EventGenerator/RES-EM">     genie::ReinSehgalRESPXSec/EM-NoPauliBlock     </param>

<param type="alg" name="XSecModel@genie::EventGenerator/DIS-CC">     genie::KNOTunedQPMDISPXSec/Default     </param>
<param type="alg" name="XSecModel@genie::EventGenerator/DIS-NC">     genie::KNOTunedQPMDISPXSec/Default     </param>
<param type="alg" name="XSecModel@genie::EventGenerator/DIS-EM">     genie::KNOTunedQPMDISPXSec/Default     </param>
```

```
<param type="alg" name="XSecModel@genie::EventGenerator/MEC-NC">     genie::EmpiricalMECPXSec2015/Default  </param>
<param type="alg" name="XSecModel@genie::EventGenerator/MEC-EM">     genie::EmpiricalMECPXSec2015/Default  </param>
<param type="alg" name="XSecModel@genie::EventGenerator/NucleonDecay">  genie::DummyPXSec/Default     </param>
<param type="alg" name="XSecModel@genie::EventGenerator/NNBarOsc">     genie::NNBarOscDummyPXSec/Default     </param>
```

Tune in GENIE

- For each tune, there is a *$GENIE/config/G18_01a/ModelConfiguration.xml*
- All the physics models used for this tune are described in this xml file.




```xml
<param_set name="Default">

   <param type="string" name="CommonParam"> CKM </param>

   <param type="double" name="QEL-CC-XSecScale"> 1.000 </param>

   <param type="alg"  name="XSec-Integrator"> genie::NewQELXSec/Default </param>

   <param type="alg"  name="IntegralNuclearModel"> genie::NuclearModelMap/Default </param>
   <param type="string" name="IntegralNucleonBindingMode"> UseNuclearModel </param>
   <param type="double" name = "IntegralNuclearInfluenceCutoffEnergy"> 10.0 </param>

   <param type="alg" name = "PauliBlockerAlg"> genie::PauliBlocker/Default </param>
   <param type="bool" name = "DoPauliBlocking"> true </param>

</param_set>

<param_set name="Dipole">
   <param type="alg"  name="FormFactorsAlg"> genie::LwlynSmithFFCC/Dipole </param>
</param_set>
```

Tune in GENIE

- For each tune, there is a *$GENIE/config/G18_01a/ModelConfiguration.xml*
- All the physics models used for this tune are described in this xml file.

Lets focus on QEL-CC process only

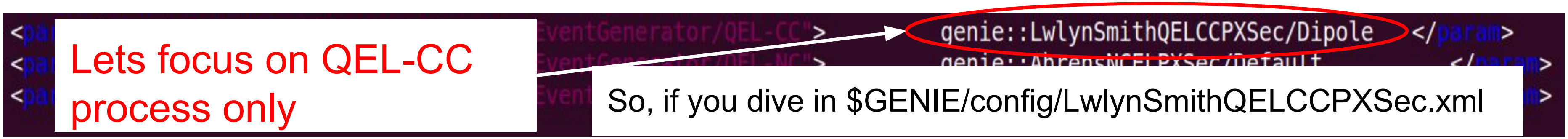
```
EventGenerator/QEL-CC">    genie::LwlynSmithQELCCPXSec/Dipole  </param>
```

So, if you dive in $GENIE/config/LwlynSmithQELCCPXSec.xml

```
<param_set name="Default">

   <param type="string" name="CommonParam"> CKM </param>

  <param_set name="HistoricalFit">
    <param type="double" name="QEL-Ma"> 0.990 </param>
    <param type="double" name="QEL-FA0"> -1.2670 </param>
  </param_set>

</param_set>

<param_set name="Dipole">
   <param type="alg"  name="FormFactorsAlg"> genie::LwlynSmithFFCC/Dipole </param>
</param_set>
```

Tune in GENIE

- For each tune, there is a *$GENIE/config/G18_01a/ModelConfiguration.xml*
- All the physics models used for this tune are described in this xml file.




```
<param_set name="HistoricalFit">
  <param type="double" name="QEL-Ma"> 0.990 </param>
  <param type="double" name="QEL-FA0"> -1.2670 </param>
</param_set>
```

- You can change these parameter’s value and **tune with data** or you can create such a xml file to **test your own model**.
- But you should be extremely careful of what you are doing!!!

Event Generator List

- What kind of interaction events you can generate?

=> Look at *$GENIE/config/EventGeneratorListAssembler.xml* – you will find whole bunch of process that you can generate

You can generate **Quasi-Elastic (QE)** events only or **Charge Current Quasi-Elastic (CCQE)** events

```
<param_set name="QE">
   <param type="int" name="NGenerators">   2                             </param>
   <param type="alg" name="Generator-0">   genie::EventGenerator/QEL-CC  </param>
   <param type="alg" name="Generator-1">   genie::EventGenerator/QEL-NC  </param>
</param_set>

<param_set name="CCQE">
   <param type="int" name="NGenerators">   1                             </param>
   <param type="alg" name="Generator-0">   genie::EventGenerator/QEL-CC  </param>
</param_set>
```

Event Generator List

- What kind of interaction events you can generate?

=> Look at *$GENIE/config/EventGeneratorListAssembler.xml* – you will find whole bunch of process that you can generate

You can generate **DIS** events only or **Charge Current DIS (CCDIS)** events or **Neutral-Current DIS (NCDIS)** events

```
<param_set name="DIS">
   <param type="int" name="NGenerators">     2                                   </param>
   <param type="alg" name="Generator-0 ">    genie::EventGenerator/DIS-CC        </param>
   <param type="alg" name="Generator-1 ">    genie::EventGenerator/DIS-NC        </param>
</param_set>

<param_set name="CCDIS">
   <param type="int" name="NGenerators">     1                                   </param>
   <param type="alg" name="Generator-0 ">    genie::EventGenerator/DIS-CC        </param>
</param_set>

<param_set name="NCDIS">
   <param type="int" name="NGenerators">     1                                   </param>
   <param type="alg" name="Generator-0">     genie::EventGenerator/DIS-NC        </param>
</param_set>
```

Event Generator List

- What kind of interaction events you can generate?

=> Look at *$GENIE/config/EventGeneratorListAssembler.xml* – you will find whole bunch of processes that you can generate

You can generate **CC** events only or **NC** events only

```
<param_set name="NC">
   <param type="int" name="NGenerators">   6                                  </param>
   <param type="alg" name="Generator-0">   genie::EventGenerator/QEL-NC       </param>
   <param type="alg" name="Generator-1">   genie::EventGenerator/RES-NC       </param>
   <param type="alg" name="Generator-2">   genie::EventGenerator/DIS-NC       </param>
   <param type="alg" name="Generator-3">   genie::EventGenerator/COH-NC-PION  </param>
   <param type="alg" name="Generator-4">   genie::EventGenerator/MEC-NC       </param>
   <param type="alg" name="Generator-5">   genie::EventGenerator/DFR-NC       </param>
</param_set>

<param_set name="CC">
   <param type="int" name="NGenerators">   9                                  </param>
   <param type="alg" name="Generator-0">   genie::EventGenerator/QEL-CC       </param>
   <param type="alg" name="Generator-1">   genie::EventGenerator/RES-CC       </param>
   <param type="alg" name="Generator-2">   genie::EventGenerator/DIS-CC       </param>
   <param type="alg" name="Generator-3">   genie::EventGenerator/COH-CC-PION  </param>
   <param type="alg" name="Generator-4">   genie::EventGenerator/DIS-CC-CHARM </param>
   <param type="alg" name="Generator-5">   genie::EventGenerator/QEL-CC-CHARM </param>
   <param type="alg" name="Generator-6">   genie::EventGenerator/MEC-CC       </param>
   <param type="alg" name="Generator-7">   genie::EventGenerator/DFR-CC       </param>
   <param type="alg" name="Generator-8">   genie::EventGenerator/QEL-CC-LAMBDA </param>
</param_set>
```

Step 3: Generate First Neutrino Event

```
$ gevgen [-h] [-r run#] -n nev -p neutrino_pdg -t target_pdg -e energy [-f flux]
        [-w] [-seed random_number_seed] [--cross-sections xml_file] [--tune
        EX00_00a_00_000] [--event-generator-list list_name]
        [--message-thresholds xml_file] [--unphysical-event-mask mask]
        [--event-record-print-level level] [--mc-job-status-refresh-rate rate]
        [--cache-file root_file]
```

Let's say we want to generate 1000 events in which ν_μ collides with ^{12}C nucleus. The ν_μ beam flux is constant over the range of 1-5 GeV energy.

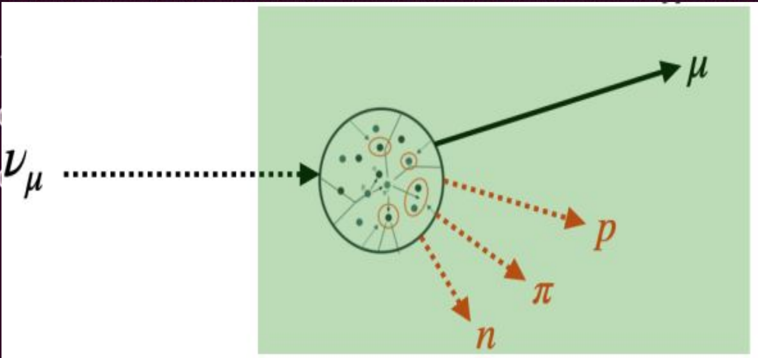
```
$ gevgen -n 1000 -e 1,5 -f 1.0 -p 14 -t 1000060120
--cross-sections /opt/mywork/gxspl-NUsmall.xml --seed 15781
--tune G18_02a_00_000 -o gntp.0.ghep.1000.root
```

Two files will be created: ***gntp.0.ghep.1000.status*** and ***gntp.0.ghep.1000.root***

Understanding GENIE event

If you open ***gntp.0.ghep.1000.status***, you will see information for 950'th event:

```
|--------------------------------------------------------------------------------------|
|GENIE GHEP Event Record [print level:   3]                                            |
|--------------------------------------------------------------------------------------|
| Idx |        Name | Ist |                                   Py |     Pz |      E |      m |
|--------------------------------------------------------------------------------------|
|  0 |        nu_mu |  0 |                                    00 |  3.775 |  3.775 |  0.000 |
|  1 |          C12 |  0 | 10                                 00 |  0.000 | 11.175 | 11.175 |
|  2 |      neutron | 11 |                                    50 | -0.005 |  0.919 |**0.940 |
|  3 |          C11 |  2 | 10                                 50 |  0.005 | 10.256 | 10.254 |
|  4 |          mu- |  1 |                                    57 |  1.828 |  2.100 |  0.106 |
|  5 |       Delta+ |  3 |                                    07 |  1.943 |  2.594 |**1.232 |
|  6 |       proton | 14 |                                    45 |  1.327 |  1.659 |  0.938 |
|  7 |          pi0 | 14 |                                    63 |  0.615 |  0.935 |  0.135 |
|  8 |       proton |  1 |                                    45 |  1.327 |  1.659 |  0.938 |
|  9 |          pi0 |  1 |        111 |  7 | -1 | -1 | -1 | 0.198 | 0.663 |  0.615 |  0.935 |  0.135 |
| 10 |      HadrBl                                                 |  0.005 | 10.256 |**0.000 |
|--------------------------------------------------------------------------------------|
|      Fin-Init:                                                  | -0.000 |  0.000 |
|--------------------------------------------------------------------------------------|
|      Vertex:        nu_mu @ (x =    0.00000 m,   =    0.00000 m, z =    0.00000 m, t =   0.000000e+00 s) |
|--------------------------------------------------------------------------------------|
| Err flag [bits:15->0] : 0000000000000000    | 1st se :                          none |
| Err mask [bits:15->0] : 1111111111111111    | Is unph sical:    NO |  Accepted:  YES  |
|--------------------------------------------------------------------------------------|
| sig(Ev) =      1.68821e-38 cm^2  | d2sig(W,Q2;E)/dWdQ2 =     1.15366e-39 cm^2/GeV^3 | Weight =        1.0000 |
|--------------------------------------------------------------------------------------|
```



Cross-section used for the event generation

Understanding GENIE event

If you open ***gntp.0.ghep.1000.status***, you will see information for 950'th event:

GENIE GHEP Event Record [print level: 3]

Idx	Name	Ist	PDG	Mother		Daughter		Px	Py	Pz	E	m
0	nu_mu	0	14	-1	-1	4	4	0.000	0.000	3.775	3.775	0.000
1	C12	0	1000060120	-1	-1	2	3	0.000	0.000	0.000	11.175	11.175
2	neutron	11	2112	1	-1	5	5	0.119	-0.150	-0.005	0.919	**0.940
3	C11	2	1000060110	1	-1	10	10	-0.119	0.150	0.005	10.256	10.254
4	mu-	1	13	0	-1	-1	-1	-0.377	-0.957	1.828	2.100	0.106
5	Delta+	3	2214	2	-1	6	7	0.496	0.807	1.943	2.594	**1.232
6	proton	14	2212	5	-1	8	8	0.298	0.145	1.327	1.659	0.938
7	pi0	14	111	5	-1	9	9	0.198	0.663	0.615	0.935	0.135
8	proton	1	2212	6	-1	-1	-1	0.298	0.145	1.327	1.659	0.938
9	pi0	1	111	7	-1	-1	-1	0.198	0.663	0.615	0.935	0.135
10	HadrBlob	15	2000000002	3	-1	-1	-1	-0.119	0.150	0.005	10.256	**0.000
Fin-Init:								0.000	0.000	-0.000	0.000	

Vertex: ... +00 s)

Err flag [bits:15-> ... none

Err mask [bits:15-> ...

sig(Ev) = 1.6 ... 1.00000

Incoming, outgoing and intermediate particle information:
PDG, momentum, energy, mass, …

Understanding GENIE event

If you open ***gntp.0.ghep.1000.status***, you will see information for 950'th event:

Incoming particles

Outgoing particles

rint level: 3]

Idx	Name	Ist	PDG	Mother		Daughter		Px	Py	Pz	E	m
0	nu_mu	0	14	-1	-1	4	4	0.000	0.000	3.775	3.775	0.000
1	C12	0	1000060120	-1	-1	2	3	0.000	0.000	0.000	11.175	11.175
2	neutron	11	2112	1	-1	5	5	0.119	-0.150	-0.005	0.919	**0.940
3	C11	2	1000060110	1	-1	10	10	-0.119	0.150	0.005	10.256	10.254
4	mu-	1	13	0	-1	-1	-1	-0.377	-0.957	1.828	2.100	0.106
5	Delta+	3	2214	2	-1	6	7	0.496	0.807	1.943	2.594	**1.232
6	proton	14	2212	5	-1	8	8	0.298	0.145	1.327	1.659	0.938
7	pi0	14	111	5	-1	9	9	0.198	0.663	0.615	0.935	0.135
8	proton	1	2212	6	-1	-1	-1	0.298	0.145	1.327	1.659	0.938
9	pi0	1	111	7	-1	-1	-1	0.198	0.663	0.615	0.935	0.135
10	HadrBlob	15	2000000002	3	-1	-1	-1	-0.119	0.150	0.005	10.256	**0.000
Fin-Init:								0.000	0.000	-0.000	0.000	

Vertex: ... +00 s)

Err flag [bits:15-> ... none

Err mask [bits:15->

sig(Ev) = 1.6 ... 1.00000

Incoming, outgoing and intermediate particle information:
PDG, momentum, energy, mass, …

Understanding GENIE event

If you open ***gntp.0.ghep.1000.status***, you will see information for 950'th event:

Incoming particles

Outgoing particles

```
print level:   3]
```

Idx	Name	Ist	PDG	Mother		Daughter		Px	Py	Pz	E	m
0	nu_mu	0	14	-1	-1	4	4	0.000	0.000	3.775	3.775	0.000
1	C12	0	1000060120	-1	-1	2	3	0.000	0.000	0.000	11.175	11.175
2	neutron	11	2112	1	-1	5	5	0.119	-0.150	-0.005	0.919	**0.940
3	C11	2	1000060110	1	-1	10	10	[illegible]	[illegible]	[illegible]	[illegible]	.254
4	mu-	1	13	0	-1	-1	-1	[illegible]	[illegible]	[illegible]	[illegible]	.106
5	Delta+	3	2214	2	-1	6	7	[illegible]	[illegible]	[illegible]	[illegible]	.232
6	proton	14	2212	5	-1	8	8	0.298	0.145	1.327	1.659	0.938
7	pi0	14	111	5	-1	9	9	0.198	0.663	0.615	0.935	0.135
8	proton	1	2212	6	-1	-1	-1	0.298	0.145	1.327	1.659	0.938
9	pi0	1	111	7	-1	-1	-1	0.198	0.663	0.615	0.935	0.135
10	HadrBlob	15	2000000002	3	-1	-1	-1	-0.119	0.150	0.005	10.256	**0.000

RES CC: $\nu_\mu + n \rightarrow \mu^- + p + \pi^0$

```
Fin-Init:        |  0.000 |  0.000 | -0.000 |  0.000 |
Vertex:          nu_mu @ (x = 0.00000 m, y = 0.00000 m, z = 0.00000 m, t = 0.000000e+00 s)
Err flag [bits:15->0 ...                                        none
Err mask [bits:15->0 ...
sig(Ev) =        1.68 ...                                       1.00000
```

Incoming, outgoing and intermediate particle information:
PDG, momentum, energy, mass, …

Understanding GENIE event

If you open ***gntp.0.ghep.1000.status***, you will see information for 950’th event:

```
|---------------------------------------------------------------------
|GENIE GHEP Event Record [print level:   3]
|---------------------------------------------------------------------
| Idx |         Name | Ist |        PDG |   Mother   | Daughter
|---------------------------------------------------------------------
|   0 |        nu_mu |   0 |         14 |  -1 |  -1 |   4 |
|   1 |          C12 |   0 | 1000060120 |  -1 |  -1 |   2 |
|   2 |      neutron |  11 |       2112 |   1 |  -1 |   5 |
|   3 |          C11 |   2 | 1000060110 |   1 |  -1 |  10 |
|   4 |          mu- |   1 |         13 |   0 |  -1 |  -1 |
|   5 |       Delta+ |   3 |       2214 |   2 |  -1 |   6 |
|   6 |       proton |  14 |       2212 |   5 |  -1 |   8 |
|   7 |          pi0 |  14 |        111 |   5 |  -1 |   9 |
|   8 |       proton |   1 |       2212 |   6 |  -1 |  -1 |
|   9 |          pi0 |   1 |        111 |   7 |  -1 |  -1 |
|  10 |     HadrBlob |  15 | 2000000002 |   3 |  -1 |  -1 |
|---------------------------------------------------------------------
|      Fin-Init:
|---------------------------------------------------------------------
|      Vertex:        nu_mu @ (x =     0.00000 m, y =     0.
|---------------------------------------------------------------------
| Err flag [bits:15->0] : 0000000000000000    |  1st set:
| Err mask [bits:15->0] : 1111111111111111    |  Is unphysical:
|---------------------------------------------------------------------
| sig(Ev) =       1.68821e-38 cm^2  | d2sig(W,Q2;E)/dWdQ2 =
|---------------------------------------------------------------------
```

Description	Status
Initial State	0
Stable final state	1
Intermediate state	2
Decayed state	3
Nucleon target	11
DIS pre-frag hadronic state	12
Resonant pre-decayed state	13
Hadron in the nucleus	14
Remnant nucleus	15
Undefined state	-1

Understanding GENIE event

```
------------------------------------------------------------------------------------------
GENIE Interaction Summary
------------------------------------------------------------------------------------------
[-] [Init-State]
 |--> probe          : PDG-code = 14 (nu_mu)
 |--> nucl. target   : Z = 6, A = 12, PDG-Code = 1000060120 (C12)
 |--> hit nucleon    : PDC-Code = 2112 (neutron)
 |--> hit quark      : no set
 |--> probe 4P       : (E =     3.775364, Px =     0.000000, Py =     0.000000, Pz =     3.775364)
 |--> target 4P      : (E =    11.174863, Px =     0.000000, Py =     0.000000, Pz =     0.000000)
 |--> nucleon 4P     : (E =     0.919068, Px =     0.118540, Py =    -0.149505, Pz =    -0.004542)
[-] [Process-Info]
 |--> Interaction : Weak[CC]
 |--> Scattering  : RES
[-] [Kinematics]
 |--> *Selected* Bjorken x = 0.620209
 |--> *Selected* Inelasticity y = 0.472252
 |--> *Selected* Momentum transfer Q2 (>0) = 2.042625
 |--> *Selected* Hadronic invariant mass W = 1.434950
[-] [Exclusive Process Info]
 |--> charm prod.    : false |--> strange prod.  : false
 |--> f/s nucleons : N(p) = 0 N(n) = 0
 |--> f/s pions      : N(pi^0) = 0 N(pi^+) = 0 N(pi^-) = 0
 |--> f/s Other      : N(gamma) = 0 N(Rho^0) = 0 N(Rho^+) = 0 N(Rho^-) = 0
 |--> resonance      : P33(1232)
 |--> final quark prod.  : false
 |--> final lepton prod.  : false
------------------------------------------------------------------------------------------
```

Initial State Information

What kind of interaction it is?

Kinematics parameter for the event

Understanding GENIE event

But if you want to see any other event's info (for e.g. 105'th event):

```
$ gevdump -f gntp.0.ghep.1000.root -n 105
```

Incoming particles

Idx	Name	Ist	PDG	Mother		Daughter		Px	Py	Pz	E	m
0	nu_mu	0	14	-1	-1	4	4	0.000	0.000	4.080	4.080	0.0
1	C12	0	1000060120	-1	-1	2	3	0.000	0.000	0.000	11.175	11.1
2	neutron	11	2112	1	-1	5	5	-0.058	[illegible]	[illegible]	[illegible]	**0.9
3	C11	2	1000060110	1	-1	7	7	0.058	[illegible]	[illegible]	[illegible]	10.2
4	mu-	1	13	0	-1	-1	-1	-0.572	[illegible]	[illegible]	[illegible]	0.1
5	proton	14	2212	2	-1	6	6	0.514	-0.247	0.451	1.187	0.9
6	proton	1	2212	5	-1	-1	-1	0.514	-0.247	0.451	1.187	0.9
7	HadrBlob	15	2000000002	3	-1	-1	-1	0.058	0.132	-0.130	10.262	**0.0

Fin-Init: 0.000 | 0.000 | -0.000 | -0.000 |

Vertex: nu_mu @ (x = 0.00000 m, y 0.00000 m, t = 0.000000e+00 s

r flag [bits:15->0]

r mask [bits:15->0]

g(Ev) = 4.9076 1.000

QES CC: $\nu_{\mu} + n \rightarrow \mu^{-} + p$

Outgoing particles

Incoming, outgoing and intermediate particle information: PDG, momentum, energy, mass, …

Understanding GENIE event

```
$ genie
$ using namespace genie;
$ TFile *myFile = new TFile("gntp.0.ghep.1000.root");
$ TTree *myTree = dynamic_cast<TTree*>(myFile->Get("gtree"));
$ myTree->GetEntries()    //Number of events
(long long) 1000
```

Understanding GENIE event

```
$ genie
$ using namespace genie;
$ TFile *myFile = new TFile("gntp.0.ghep.1000.root");
$ TTree *myTree = dynamic_cast<TTree*>(myFile->Get("gtree"));
$ myTree->GetEntries()   //Number of events
(long long) 1000
$ genie::NtpMCEventRecord* myEventRecord = new NtpMCEventRecord();
$ myTree->SetBranchAddress("gmcrec", &myEventRecord);
$ myTree->GetEntry(0);  // Accessing first event
$ genie::EventRecord* myEvent = myEventRecord->event;
$ myEvent->XSec()
(double) 1.4123255e-10
```

Correct cross-section?

Understanding GENIE event

```
$ genie
$ using namespace genie;
$ TFile *myFile = new TFile("gntp.0.ghep.1000.root");
$ TTree *myTree = dynamic_cast<TTree*>(myFile->Get("gtree"));
$ myTree->GetEntries()    //Number of events
(long long) 1000
$ genie::NtpMCEventRecord* myEventRecord = new NtpMCEventRecord();
$ myTree->SetBranchAddress("gmcrec", &myEventRecord);
$ myTree->GetEntry(0);  // Accessing first event
$ genie::EventRecord* myEvent = myEventRecord->event;
$ myEvent->XSec()
(double) 1.4123255e-10
```

- Expressed in natural units
- To convert it into cm^{-2}, divide by (5.07e+13 * 5.07e+13)

Understanding GENIE event

```
$ myEvent->Summary()->ProcInfo().IsQuasiElastic()
(bool) false
$  myEvent->Summary()->ProcInfo().IsDeepInelastic()
(bool) true
$ myEvent->Summary()->ProcInfo().IsWeakCC()
(bool) true
```

Convert genie output into normal root file

```
$ genie 'read_genie_convert_root.cc("gntp.0.ghep.1000.root")'
```

- This will create a new root file with named "***..._converted.root***". This is a standard root files and can be analysed with root only, we won't need genie classes anymore.
- It has two trees:
  - **Event:** stored all event information, such as incoming neutrino energy, momentum, event cross-section, what kind of process it is (qe, dis, cc, nc etc)

```
$ root -l gntp.0.ghep.1000_converted.root
root[1]$ .ls
```

Project 2

Let’s look at some of the kinematics variable:

Energy transfer: $\boxed{\omega = E_\nu - E_\ell}$

It tells you about nuclear effects: Fermi motion, binding energy, FSI…

Four Momentum Transfer: $\boxed{Q^2 = -(p_\nu - p_\ell)^2}$

Determines whether the event is quasi-elastic or deep inelastic..

Three Momentum Transfer: $\boxed{|\vec{q}| = |\vec{p}_\nu - \vec{p}_\ell|}$

Controls the spatial scale of the interaction (how deeply the probe goes into the nucleus)

Bjorken x and y: $\boxed{y = \frac{\omega}{E_\nu}, \quad x = \frac{Q^2}{2m_N\omega}}$

Bjorken x: Measures the fraction of nucleon momentum carried by the struck quark– x~1: quasi-elastic, x<1: resonance or DIS, x>1: nuclear effects

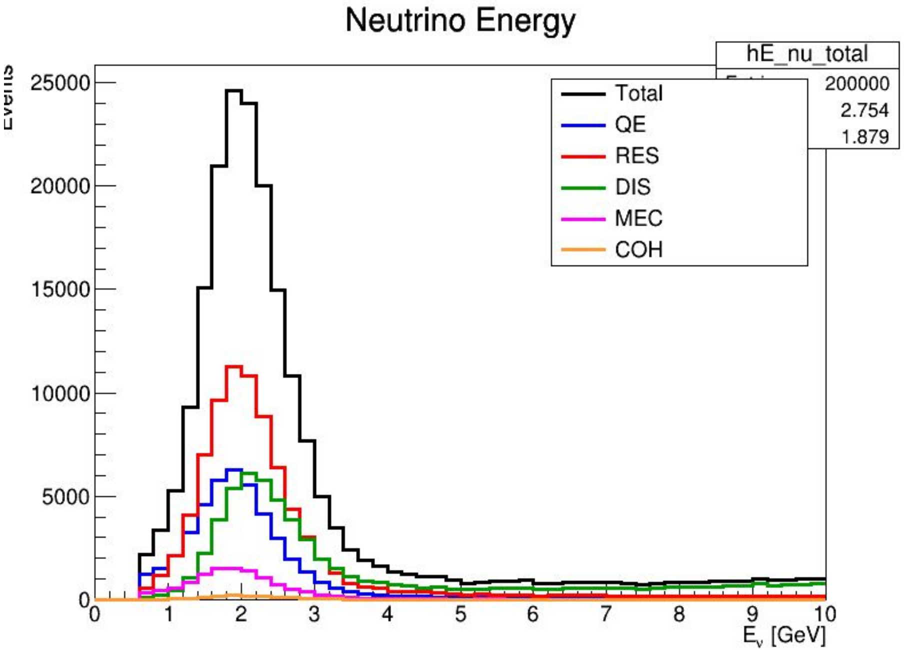
Bjroken y: Fraction of neutrino energy transferred to the target system

So, let’s generate 0.2M events with NOvA Flux with energy range of 0.2 to 10 GeV

Project 2

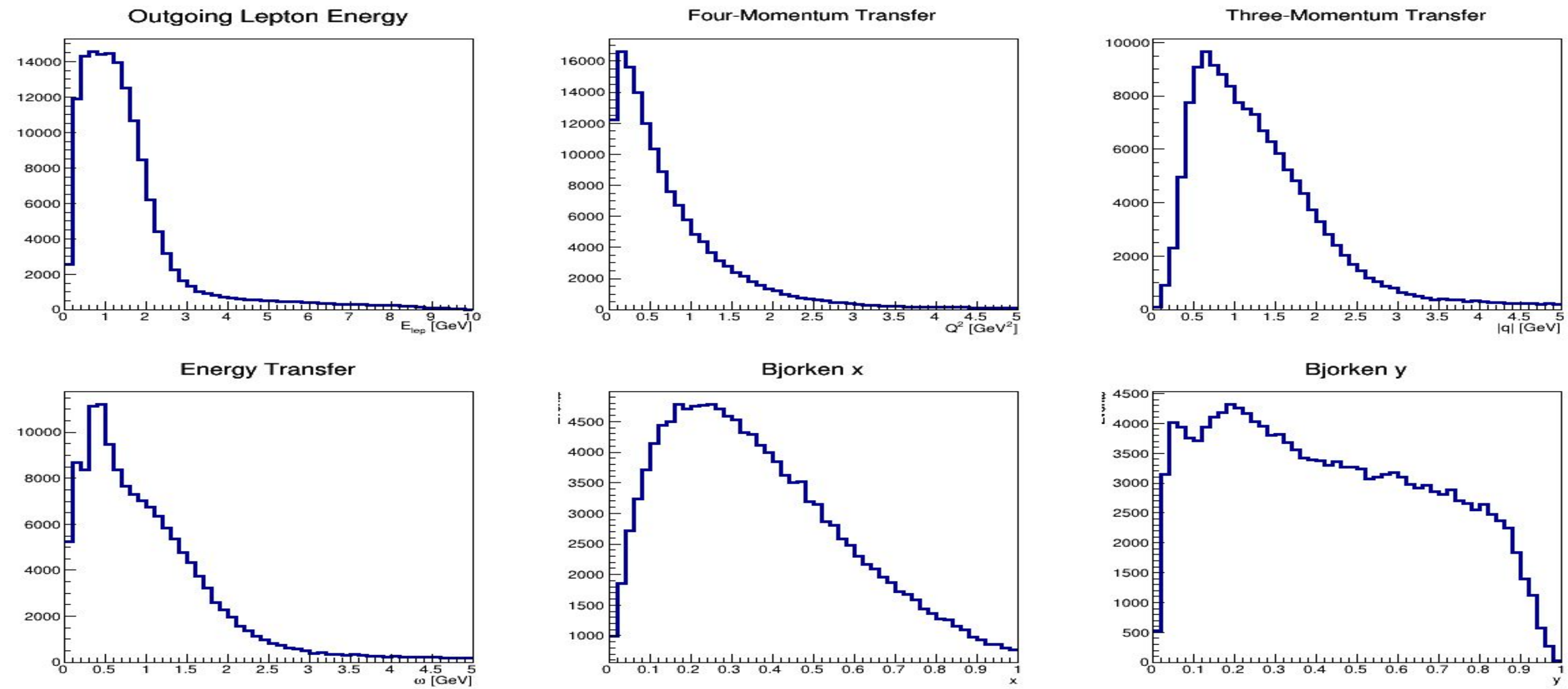
```
$ gevgen -n 200000 -p 14 -t 1000060120 -f
FHC_Flux_NOvA_ND_2017.root,flux_numu -e 0.2,10 -r 1000 --seed
292929 --cross-sections /opt/mywork/gxspl-NUsmall.xml --tune
G18_02a_00_000 -o truth.ghep.root
```

You can generate this.
But it takes some time, so I have provided you a root file which is converted from genie output.
You can use that file:
`truth.ghep_converted.root`
From this file, we will look into some kinematic variables..



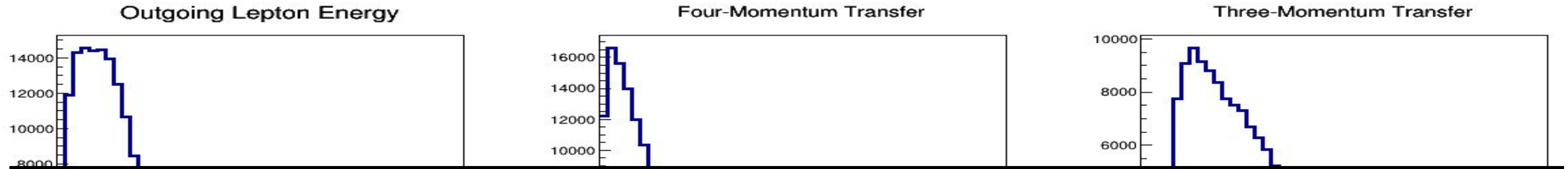
Project 2

```
$ root -l
'plot_genie_kinematics.cc("../truth.ghep_converted.root")'
```



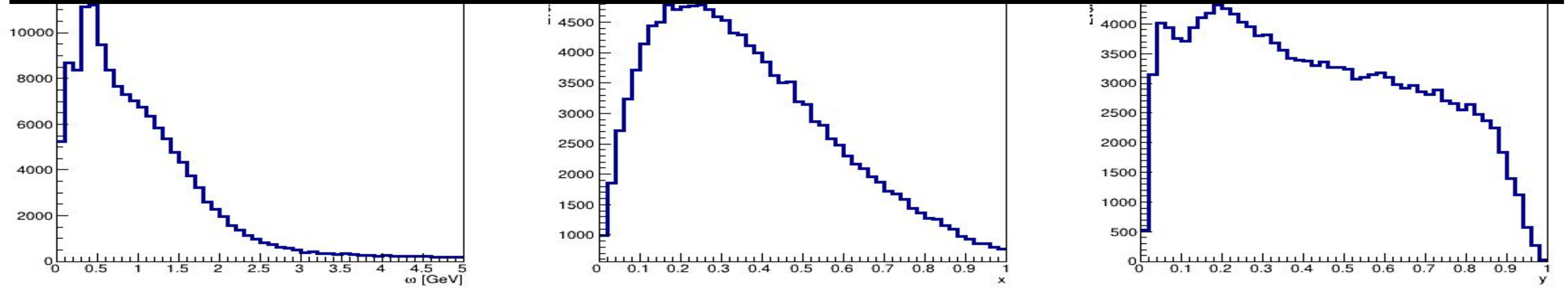
Project 2

```
$ root -l
'plot_genie_kinematics.cc("../truth.ghep_converted.root")'
```



Can you plot the angle of the outgoing lepton?

$$\cos\theta_\ell = \frac{p_z}{|\vec{p}_\ell|}$$



Project 3

- We have heard a lot about three flavor oscillations.
- Now let's say we have an energy spectrum at the ND (generated from GENIE)
- So, we can use oscillation formula and see how the spectra changes.

Vacuum oscillation probability:

$$P^{\text{vac}}_{\nu_\mu \to \nu_\mu} \simeq 1 - \cos^4\theta_{13} \sin^2(2\theta_{23}) \sin^2\left(1.267 \frac{\Delta L}{E}\right)$$

Matter oscillation probability:

$$P^{\text{matter approx}}_{\nu_\mu \to \nu_\mu} \simeq 1 - \cos^4\theta^m_{13} \sin^2(2\theta_{23}) \sin^2\left(1.267 \frac{\Delta m^2_m L}{E}\right)$$

```
$ root -l 'osc_approx_matter.cc("../truth.ghep_converted.root")'
```

Project 3

```
$ root -l 'osc_approx_matter.cc("../truth.ghep_converted.root")'
```

$$\theta_{12} = 33.44^\circ, \quad \sin^2\theta_{12} \approx 0.3037$$

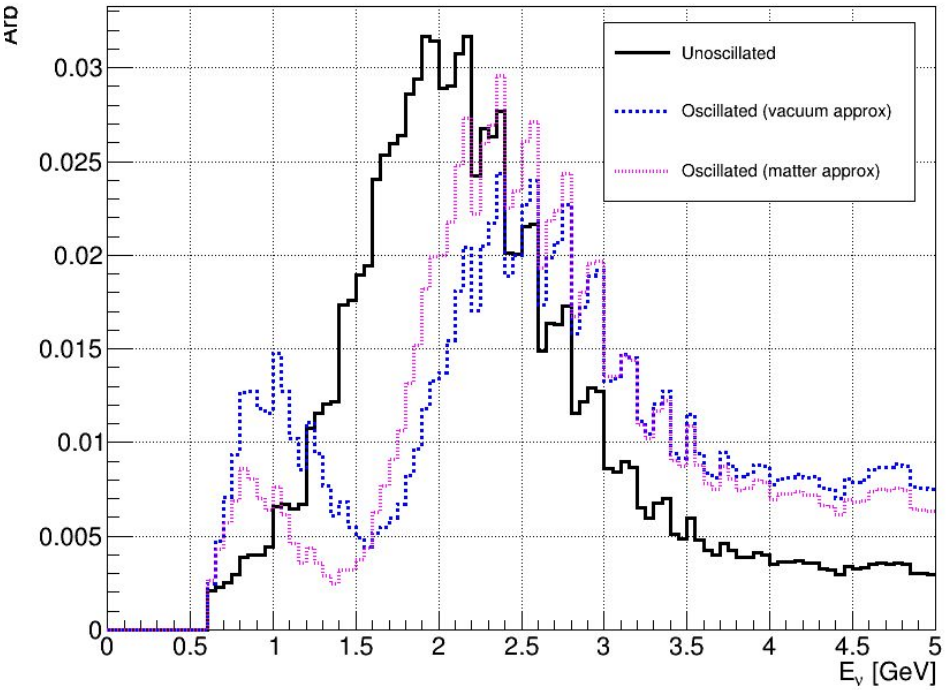
$$\theta_{13} = 8.57^\circ, \quad \sin^2\theta_{13} \approx 0.0222$$

$$\theta_{23} = 49.20^\circ, \quad \sin^2\theta_{23} \approx 0.5730$$

$$\Delta m^2_{21} = 7.42 \times 10^{-5}\ \mathrm{eV}^2$$

$$\Delta m^2_{31} = 2.517 \times 10^{-3}\ \mathrm{eV}^2$$

$$\delta_{\mathrm{CP}} = 197^\circ, \quad L_{\mathrm{NOvA}} = 810\ \mathrm{km}$$



Neutrino Energy Reconstruction

- Neutrino energy is **not known**, need to reconstruct based on the **interaction products measured in the detector.**

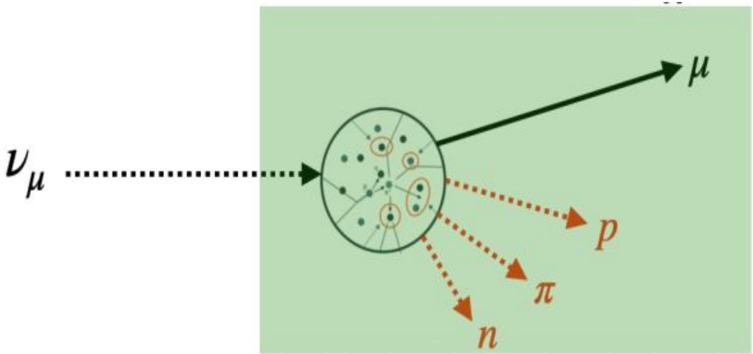
Kinematic Method

$$E_\nu^{QE} = \frac{2(M_n - E_B)E_\mu - (E_B^2 - 2M_n E_B + m_\mu^2 + \Delta M^2)}{2[(M_n - E_B) - E_\mu + p_\mu \cos\theta_\mu]}$$

- Depends on lepton reconstruction
- "CCQE" topology, selecting 1μ0π , may have contribution from 2p2h, pion absorption, etc.

Calorimetric Method

$$E_\nu^{Cal} = \sum E_{observed\ particles} + E_{neutrons} + E_{missing}$$

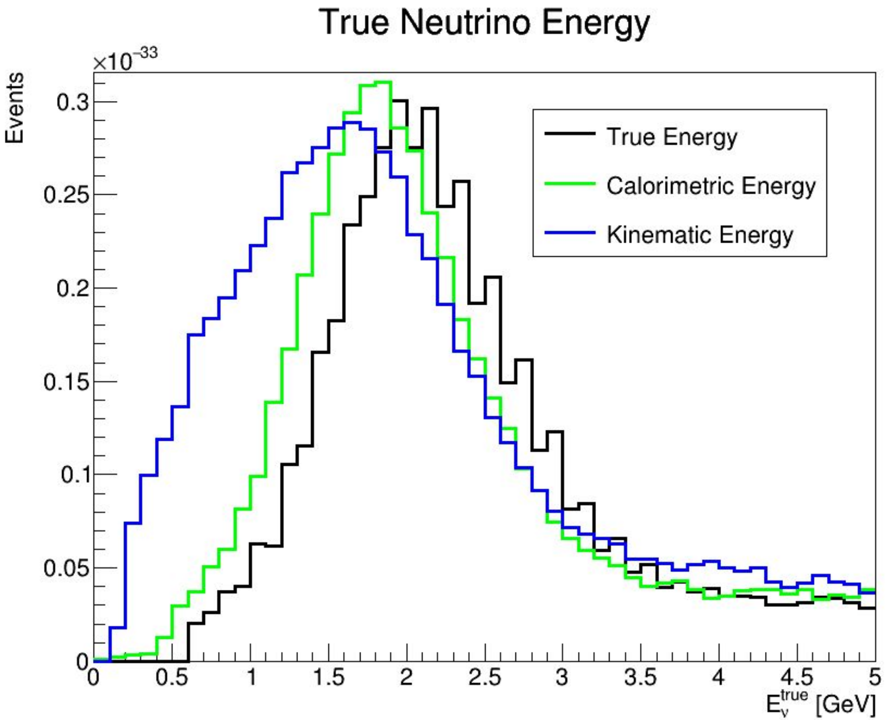


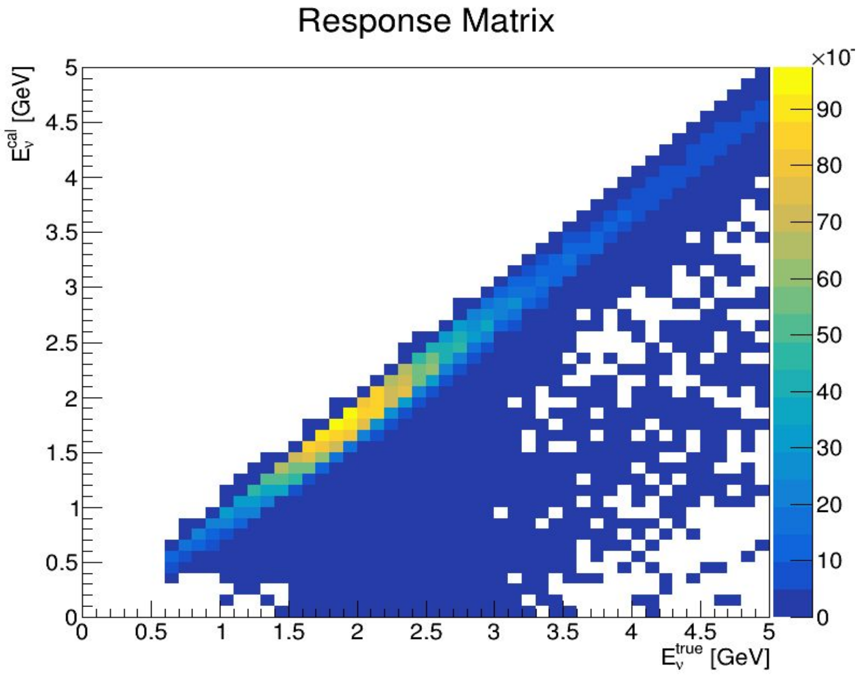
- Energy conservation, relies on visible energy
- Missing energy from neutrons, detector threshold, pion absorption, etc

In both cases, neutrino energy reconstruction requires estimates from the neutrino-nucleus interaction physics.

Project 4: Reconstruction of neutrino energy

```
$ root -l 'osc_approxroot -l
'reconstruct_energy.cc("../truth.ghep_converted.root")'
```





Some more project ideas:

- Now you know how to reconstruct energy. So, instead of using true neutrino energy, you can use either kinematic or calorimetric method to reconstruct neutrino energy and then see how the energy spectrum changes after oscillation.
- You can study Nuclear Effects:
    - Compare event generation with and without Fermi motion (e.g., free proton vs. carbon vs. argon).
    - Look at how lepton energy/angular distributions are smeared.
- You can change any model parameters value (in *modelConfiguration.xml*), (for eg. M_A) and see how does it impact the cross-section
- Non-neutrino event generation modes:
    - Boosted dark matter event generation
    - Nucleon decay
    - Neutrino-antineutrino oscillation
    - Hadron-nucleus scattering

Thank You!

Any questions? Comments?

Understanding GENIE event