How to Extract Cross Section Information from GENIE

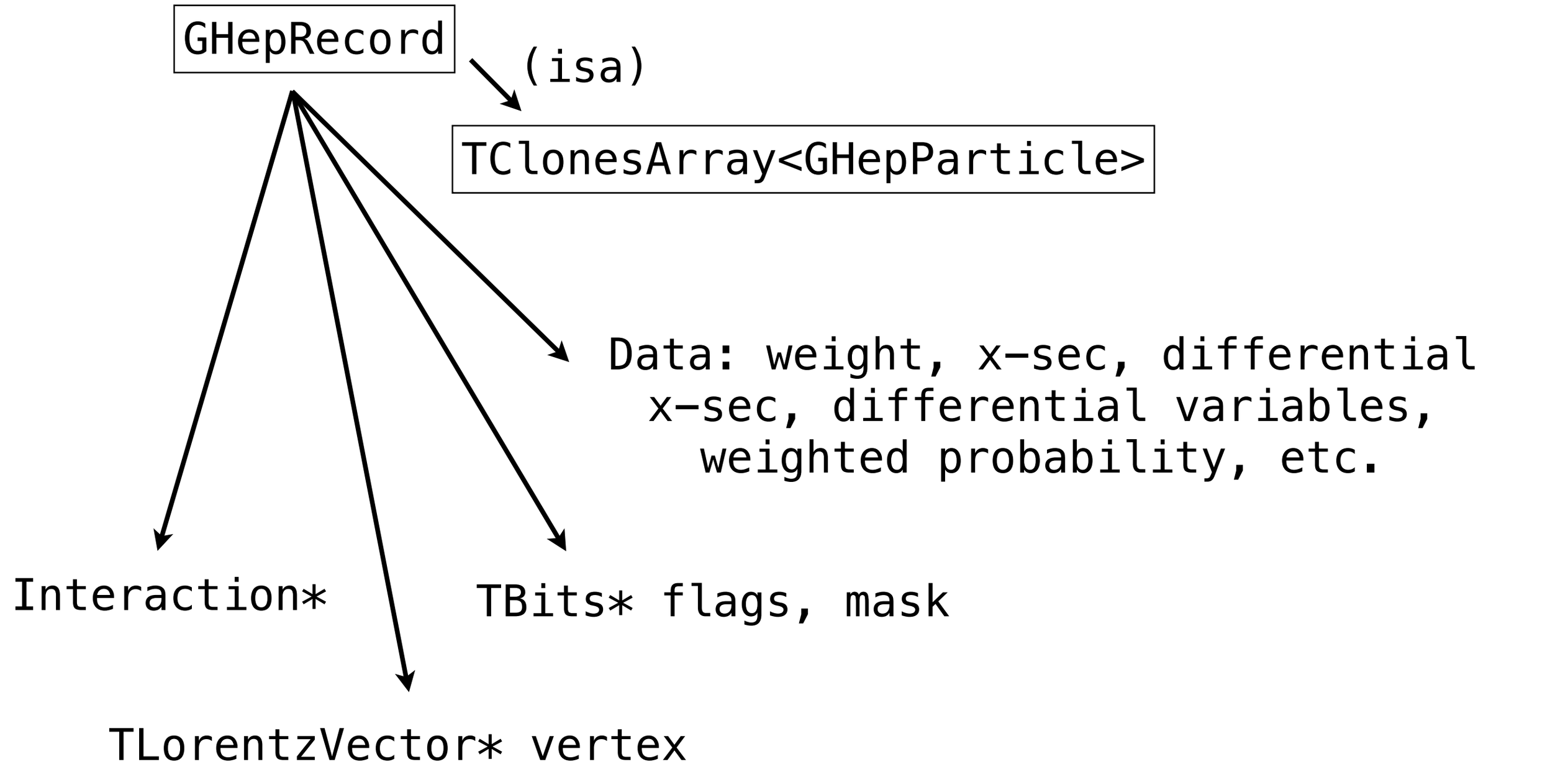
Gabriel N. Perdue
Fermilab

Special thanks to Robert Hatcher for some of the useful examples in this presentation.

Overview

- The relevant classes.
  - GHepRecord, Interaction (NtpMCEventRecord also very useful)
- The standard converter apps.
- How to analyze the outputs with ROOT.

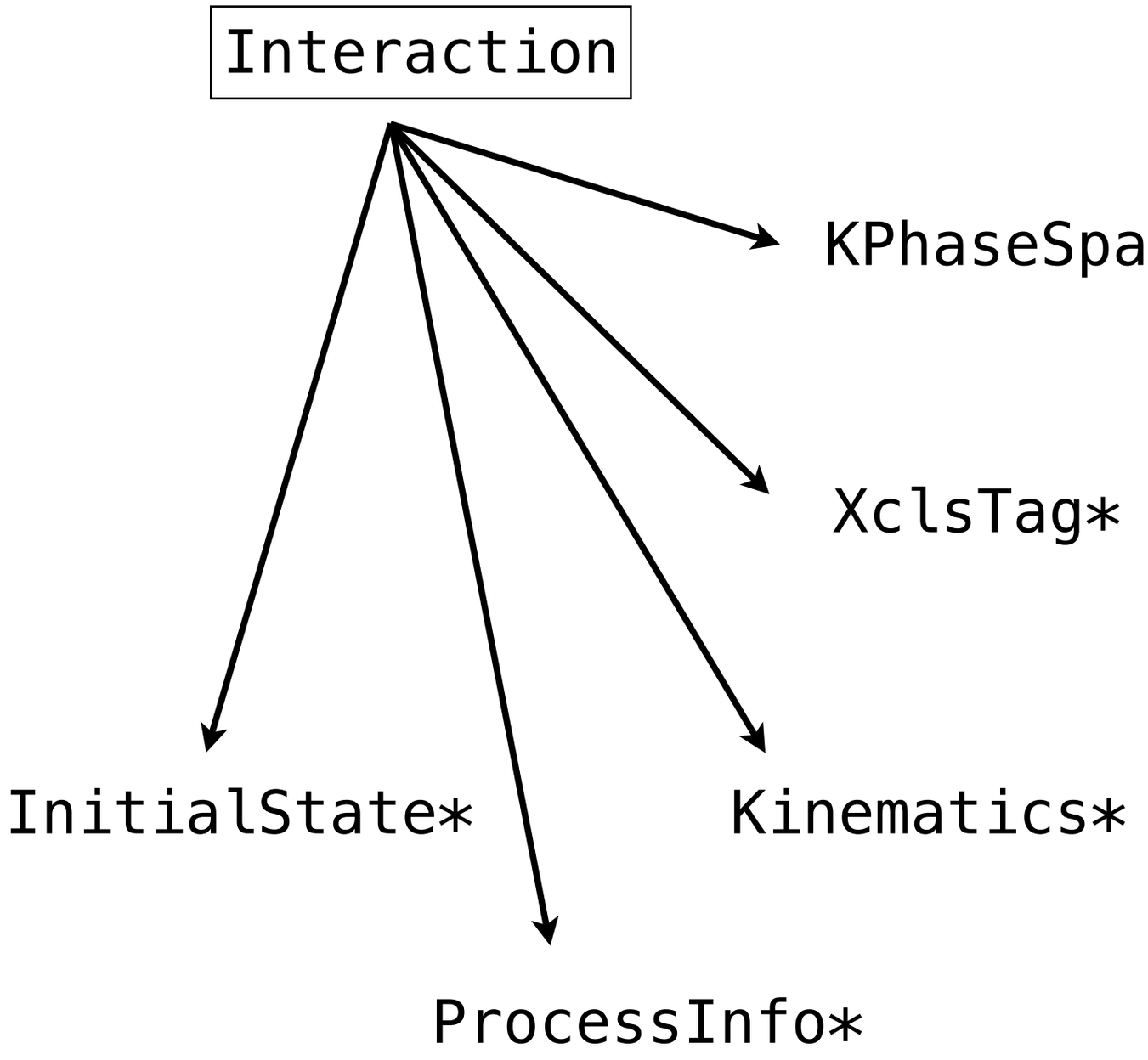
Code in $GENIE/src/GHEP:



Mother / Daughter ordering is maintained dynamically during event generation.

```
|-----------------------------------------------------------------------------------------------------------------------------------------|
|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 |   1.550 |   1.550 |   0.000 |
|   1 |           O16 |   0 | 1000080160 |  -1 |  -1 |   2 |   3 |   0.000 |   0.000 |   0.000 |  14.890 |  14.890 |
|   2 |        proton |  11 |       2212 |   1 |  -1 |   5 |   5 |  -0.090 |   0.119 |   0.114 |   0.916 | **0.938 | M = 0.897
|   3 |           N15 |   2 | 1000070150 |   1 |  -1 |  11 |  11 |   0.090 |  -0.119 |  -0.114 |  13.974 |  13.973 |
|   4 |         nu_mu |   1 |         14 |   0 |  -1 |  -1 |  -1 |  -0.463 |  -0.029 |   0.137 |   0.484 |   0.000 | P =
(0.957,0.059,-0.283)
|   5 |        Delta+ |   3 |       2214 |   2 |  -1 |   6 |   7 |   0.374 |   0.147 |   1.527 |   1.982 | **1.232 | M = 1.198
|   6 |        proton |  14 |       2212 |   5 |  -1 |   8 |   9 |   0.475 |   0.049 |   1.161 |   1.567 |   0.938 | FSI = 2
|   7 |           pi0 |  14 |        111 |   5 |  -1 |  10 |  10 |  -0.102 |   0.098 |   0.366 |   0.415 |   0.135 | FSI = 3
|   8 |       neutron |   1 |       2112 |   6 |  -1 |  -1 |  -1 |   0.791 |   0.033 |   0.499 |   1.326 |   0.940 |
|   9 |        proton |   1 |       2212 |   6 |  -1 |  -1 |  -1 |  -0.262 |  -0.078 |   0.629 |   1.162 |   0.938 |
|  10 |           pi0 |   1 |        111 |   7 |  -1 |  -1 |  -1 |  -0.115 |   0.097 |   0.362 |   0.415 |   0.135 |
|  11 |     HadrBlob |  15 | 2000000002 |   3 |  -1 |  -1 |  -1 |   0.049 |  -0.024 |  -0.078 |  13.053 | **0.000 | M = 13.052
|-----------------------------------------------------------------------------------------------------------------------------------------|
|      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 s)                        |
|-----------------------------------------------------------------------------------------------------------------------------------------|
| Err flag [bits:15->0] : 0000000000000000    |  1st set:                                                                    none |
| Err mask [bits:15->0] : 1111111111111111    |  Is unphysical:    NO |   Accepted:   YES                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------|
| sig(Ev) =       1.2945e-38 cm^2  | d2sig(W,Q2;E)/dWdQ2 =       2.399e-38 cm^2/GeV^3 | Weight =                 1 |
|-----------------------------------------------------------------------------------------------------------------------------------------|
```

Code in $GENIE/src/Interaction:



- "Lightweight" event summary.
- Many specialized methods and constructors – see the class header.
- Simple to instantiate and use to drive physics models.

Developer note – be careful how you update the GHepRecord and Interaction. There is some redundancy there & consistency is not automatically enforced.

```
------------------------------------------------------------------------------------------
GENIE Interaction Summary
------------------------------------------------------------------------------------------
[-] [Init-State]
 |--> probe        : PDG-code = 14 (nu_mu)
 |--> nucl. target : Z = 8, A = 16, PDG-Code = 1000080160 (O16)
 |--> hit nucleon  : PDC-Code = 2212 (proton)
 |--> hit quark    : no set
 |--> probe 4P     : (E =      1.55009, Px =            0, Py =            0, Pz =      1.55009)
 |--> target 4P    : (E =        14.89, Px =            0, Py =            0, Pz =           0)
 |--> nucleon 4P   : (E =     0.916096, Px =   -0.0898987, Py =     0.118759, Pz =     0.113648)
[-] [Process-Info]
 |--> Interaction : Weak[NC]
 |--> Scattering  : RES
[-] [Kinematics]
 |--> *Selected* Bjorken x = 0.630147
 |--> *Selected* Inelasticity y = 0.686746
 |--> *Selected* Momentum transfer Q2 (>0) = 1.07657
 |--> *Selected* Hadronic invariant mass W = 1.19833
[-] [Exclusive Process Info]
 |--> charm prod.  : false
 |--> f/s nucleons : N(p) = 0 N(n) = 0
 |--> f/s pions    : N(pi^0) = 0 N(pi^+) = 0 N(pi^-) = 0
 |--> resonance    : P33(1232)
------------------------------------------------------------------------------------------
```

Code in '$GENIE/src/Interaction':

- 'InitialState.{h,cxx}': target & probe data (vertices, 4-momenta, etc.)
- 'ProcessInfo.{h,cxx}': describe the physics channel (e.g., IsCoherent, etc.)
- 'Kinematics.{h,cxx}': canonical scattering variables (x,y,Q2,W,etc.)
- 'XclsTag.{h,cxx}': tags for identifying the final state (e.g., NPions in f.s., etc.)
- 'KPhaseSpace.{h,cxx}': allowed kinematic range.

StdApps

- `gevdump`: Pretty-print the GHepRecord.
- `gntpc`: Convert the native GHEP file to one of plain text, XML, or flat ROOT.
- `gspl2root`: Convert the total cross section XML spline into a ROOT format.
  - Note, you may not specify an event list here, so if not all channels are present in your XML, GENIE will recalculate them.

gevdump

```
# print the first event

$ gevdump -f gntp.0.ghep.root -n 0

# print the third event

$ gevdump -f gntp.0.ghep.root -n 2

# print all the events

$ gevdump -f gntp.0.ghep.root

# limit print level

$ gevdump -f gntp.0.ghep.root --event-record-print-level 0
```

```
// Print levels:
//  0 -> prints particle list
//  1 -> prints particle list + event flags
//  2 -> prints particle list + event flags + wght/xsec
//  3 -> prints particle list + event flags + wght/xsec + summary
// 10 -> as in level 0 but showing particle positions too
// 11 -> as in level 1 but showing particle positions too
// 12 -> as in level 2 but showing particle positions too
// 13 -> as in level 3 but showing particle positions too
```

gntpc

```
# create a flat ntuple (most popular option)

$ gntpc -i ./gntp.1.ghep.root -f gst

# create a ROOT STDHEP-like event tree

$ gntpc -i ./gntp.1.ghep.root -f rootracker
```

- These will make files like:
  - 'gntp.1.gst.root' (gst)
  - 'gntp.1.gtrac.root' (rootracker)
- (Note: here “1” is the run number, etc.)

(There are many other format options.)

Demos: Get Started

- Get the 'SimpleXSecAna' package:

```
$ git clone https://github.com/GENIEMC/SimpleXSecAna.git
$ cd SimpleXSecAna
$ . ana_setup.sh
```

- A spline file for muon neutrino CCQE events on scintillator (CH) is already present. It was created with the 'do_make_ccqe_scint_spline.sh' script. (It is CCQE only, so it is pretty fast to re-generate.)

```
$ cd data && ./install_splines.sh  # assumes XSECSPLINEDIR
$ time ./do_a_run.sh
```

- Should run in less than 30 seconds.

```
$ ./convert_files.sh
```

- Produce `gst` and `gtrac` files (we don't need them yet).

Aside: Do other runs

- If you used `lamp` to install GENIE, you have a more complete cross section file installed.
  - gxspl-vA-v2.8.0.xml
  - Targets: n, p, C, O, Ne, Al, Si, Ar, Fe
  - nu-e, nu-mu, nu-tau (? on H)
- If you didn't use `lamp`, you may download the file by hand from:
  - http://www.hepforge.org/archive/genie/data
- The run script provided in 'SimpleXSecAna' is easy to modify to use this spline file and to change the generator list.
- After looking at the QE examples in this presentation, make some samples using the default generator list and try to extract some cross section information, for example by cutting on final state particles.

Interactive GENIE

```
$ more `which genie`

#!/bin/sh
root $GENIE/src/scripts/gcint/genie.C $*
```

- Read the script – it loads the GENIE libraries, making it possible for us to run real GENIE code interactively in a ROOT session.

Interactive GENIE

```
$ genie gntp.100.ghep.root          # This will put us into a ROOT session

genie[2] TTree *mytree = 0;

genie[3] _file0->GetObject("gtree", mytree);

genie[4] Long64_t nEntries = mytree->GetEntries();

genie[5] cout << nEntries << endl;

1000

genie[6] genie::NtpMCEventRecord* myentry = new genie::NtpMCEventRecord();

... (this will produce the welcome message)

genie[7] mytree->SetBranchAddress("gmcrec", &myentry);

genie[8] mytree->GetEntry(0);

genie[9] myentry->PrintToStream(cout);

... (a lot of stuff)

genie[10] myentry->event->XSec()

(const double)1.4816455971583265e-10
```

GENIE Analysis

- Interactive sessions are useful, but not sufficient for complex analyses.

```
$ cd ../src

$ make   # GENIE must be compiled and set up
```

```
$ cd ../run

$ ./run_simpleAna.sh
```

- This will produce a lot of output using the GENIE message service.
- Let's go back and look at the code...

```
$ cd ../src && vim simpleAna.cxx   # Editor of choice
```

- All the GENIE classes are available here.

- For GENIE power users, this is a very convenient way to do analysis:

```
Interaction *in = event.Summary();
const ProcessInfo &proc = in->ProcInfo();
const XclsTag &xclsv = in->ExclTag();
bool qelcc = proc.IsQuasiElastic() && proc.IsWeakCC();
bool charm = xclsv.IsCharmEvent();

if (qelcc && !charm) {
  LOG("myAnalysis", pNOTICE) << event.XSec();
}


....

if (p->Status() == kISStableFinalState ) {
  if (p->Pdg() == kPdgPi0 ||
      p->Pdg() == kPdgPiP ||
      p->Pdg() == kPdgPiM)
  {
    LOG("myAnalysis", pNOTICE)
      << "Got a : " << p->Name() << " with E = " << p->E() << " GeV";
  }
}
```

Demos: gst

- Most experiments use the `gst` files for analysis (or a similar custom format).
- The `gst` file is a flat ntuple and the interpretation of the variables is "obvious" without referencing GENIE class documentation.
- Very useful for interactive plotting, etc. as well.

gst

- Check out, build, and "run" the 'SimpleXSecAna' package from earlier in this presentation.

```
$ cd $ANA_DATA_DIR

$ root -l gntp.100.gst.root

root [1] TBrowser tb

root [2] gst->Draw("El","Q2 < 1.0")

root [3] gst->Draw("W:Ws")

root [4] gst->Draw("nfp")
```

gst

- Also can produce analysis classes for making histograms, etc.
- Use ROOT's `MakeClass`.
- Here, you may want to re-generate the .h files (or modify them to use larger arrays where sensible) if you create new samples.

```
$ root -l gntp.100.gst.root
root [0]
Attaching file gntp.100.gst.root as _file0...
root [1] .ls
TFile**         gntp.100.gst.root
 TFile*         gntp.100.gst.root
  KEY: TTree gst;1    GENIE Summary Event Tree
root [2] gst->MakeClass("gstana");
Info in <TTreePlayer::MakeClass>: Files: gstana.h and gstana.C
generated from TTree: gst
```

gst and gRooTracker

```
$ cd $ANA_RUN_DIR

$ ./run_gstgroo.sh        # make histograms

$ cd ../hist

$ root -l qel_grooana.root

root [1] h_evtXSec->Draw()

root [2] h_evtDXSec->Draw()
```

- The `gRooTracker` format appears to be something somewhat T2K specific – however, it does by default include cross sections.
  - To make more sophisticated plots, look at the Physics and Users Manual for guidance on the other variables.

gst and gRooTracker

```
$ root -l qel_gstana.root

root [1] h_ev->Draw()

root [2] h_Q2->Draw()
```

- etc.
- Study these histogram-making classes by moving back up and over to 'src':

```
$ cd ../src

$ ls *.h *.cxx

grooana.cxx  grooana.h  gstana.cxx  gstana.h  Rungroo.cxx
Rungst.cxx  simpleAna.cxx
```

gst and gRooTracker

- The `Runxyz.cxx` files contain `main()`s and loop over all the runs in a list that is auto-generated by the 'convert_files.sh' script that exists in $ANA_DATA_DIR.
- It builds a chain and then loops the `xyz` classes over the chain.
- There is obviously more than one way to do this... use your favorite approaches!