Genie

Hands-on session

Dr Marco Roda
marco.roda@liverpool.ac.uk

12-14 June 2024 - INSS 2024
Bologna, Italy

UNIVERSITY OF LIVERPOOL

Introduction

Monte Carlo generators: fancy random generators

- Connect neutrino fluxes and observables
  - predict event topologies and kinematics
  - Somehow feel the gaps of the theory
- Experiments and analysers need more
  - Coverage of physics processes
  - Uncertainty validation against data
  - Tune against data in order to obtain
    - Optimised initial configuration
    - Data-driven constraints of the generator parameters
  - Capability to propagate configuration changes to prediction
    - Usually reweighting
  - Support for geometry and flux

Why GENIE?

- GENIE is widely used
  - all running neutrino beam experiments and most of the neutrino telescopes
    - JUNO also uses GENIE
    - It's even used at LHC in FASER
  - For most of these experiments GENIE is the main generator
    - For neutrino beams the only exception is T2K, but GENIE is used as well

- Probably you will not work directly with any generator
  - Mostly accessed via the wrappers provided by your experiment(s)
  - But chances are you will be working on MC data coming from GENIE

A few info about be

Why me for a GENIE session?

- I’m a member of
  - GENIE
  - DUNE
  - SBND
- Mostly worked in
  - Generator - tuning, model and framework development
  - DAQ software
- I’m Italian but living in Liverpool

What I expect at the end of the sessions

- Basic GENIE concept:
  - Generate splines and events
  - Understand the output of GENIE
    - what info is available and what is not available
    - Create simple scripts to analyse the output
  - Been able to navigate the configurations
    - Been able to make minor changes

- Areas we won't have time to cover
  - Uncertainty propagation, A.K.A. reweight
  - Geometry and flux drivers
  - Alternative operation modes: NNBar oscillation, HNL decay, Very High energy
  - Tuning

What I am assuming you know

- Management of a linux working system
  - GENIE is written to run on linux, although because of the low requirements it could run on mac as well
    - Although if I understand it correctly at the moment it's not
  - It's not expected to be able to being extended to other platforms
- C++
  - GENIE is written in C++ and we don't have (yet) python interfaces nor python bindings
  - Only simple C++ logic is necessary
- ROOT
  - A lot ROOT is involved in this hands-on session because GENIE output strongly rely on it's interfaces
- XML
  - Configuration from GENIE are handled in xml plus some machinery we built on top of it
  - What is required is very simple, so I don't think anyone need a particular understanding of xml

GENIE Overview

Status of GENIE

- Two main efforts
  - Model development
  - Tuning
- Contacts, details and code are all available from our website: www.genie-mc.org/
  - We have a mailing list for users that you can join
  - We also have slack which is public
- GENIE manual - https://genie-docdb.pp.rl.ac.uk/cgi-bin/ShowDocument?docid=2
  - It is a very extensive manual
- GitHub based project https://github.com/GENIE-MC
  - 2 main repositories for now
    - Generator
    - Reweight
- Latest release: version 3.04.02, released in April 2024
  - Previous release was 3.04.00, released in March 2023
  - http://releases.genie-mc.org/
- Recent publications
  - Neutrino-nucleon cross-section model tuning in GENIE v3 - Phys.Rev.D 104 (2021) 7, 072009
  - Hadronization model tuning in genie v3 - Phys.Rev.D 105 (2022) 1, 012009
  - Recent highlights from GENIE v3 - Eur.Phys.J.ST 230 (2021) 24, 4449-4467
  - Neutrino-nucleus CC0π cross-section tuning in GENIE v3 - Phys. Rev. D 106 (2022) 11, 112001
  - First combined tuning on transverse kinematic imbalance data with and without pion production constraints - Arxiv 2404.08510

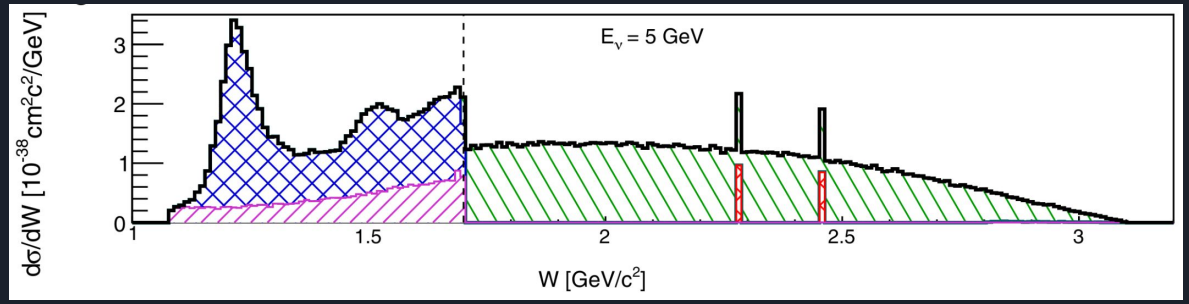
GENIE specifics - core assumptions

- As general as possible
  - GENIE can generate events with neutrino, electrons and hadrons as probes
  - On all targets - i.e. all isotopes
  - On all possible energy range
    - We have specifications for very low energies and very high energy
  - We also have some BSM processes

- Highly configurable
  - Apart from bugs, we strive to maintain backward compatibility
  - For every process we have many variations of the model
  - Consistent and valid configuration are defined through the use of tunes
    - https://hep.ph.liv.ac.uk/~costasa/genie/tunes.html

GENIE specifics - physics

- The core system does not have particular restrictions on what can be implemented
  - Maybe the only exception is polarisation inputs
- There are some areas of the physics which are GENIE specific
- Shallow inelastic region (SIS)
  - The Non-Res background is a scaled DIS evaluated in the lower W region
  - Scaling depends on final particle multiplicity



GENIE event record visitor logic

- The event generation chain is an Event Record visitor
- A block of memory (EventRecord) is created at the beginning of the chain
  - A chain is selected based on
    - Configuration
    - Precomputed quantities (splines)
  - The same block of memory is passed to different modules
    - Each one add something on the event
    - Until the event is complete
  - Ultimately the event record is the output
- Roughly, the chain of modules maps the scattering type
- The list of chains is available here
  - https://github.com/GENIE-MC/Generator/blob/master/config/EventGenerator.xml

GENIE output format - splines

- The splines are pre-calculated quantities that are used to optimise the generation
  - They take from 10 minutes to 100 hours to generate, depending what you need to do
    - That's why we distribute the most used ones
  - The main quantities are the integrated cross sections vs neutrino energy
    - for every combination of process, target and probe flavour
- They are stored in xml files
  - Divided in blocks for each tune
- They can be downloaded from https://scisoft.fnal.gov/scisoft/packages/genie_xsec/
- They can be converted in ROOT files with gspl2root
  - They become TGraphs

GENIE output formats

- The native output of GENIE is a ghep file
  - It's a ROOT Tree file
  - Tree contains the EventRecord from GENIE
    - Definition of the record
- This can be converted in a number of formats
  - Many of them are simple ROOT files that can be read by any ROOT instance without genie
  - The manual contains all the details

GENIE output format - events

```
|-----------------------------------------------------------------------------------------------------------------------|
|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 |    2.029 |    2.029 |   0.000 |
|   1 |          Ar40 |   0 | 1000180400 |  -1 |  -1 |   2 |   3 |    0.000 |    0.000 |    0.000 |   37.216 |  37.216 |
|   2 |        proton |  11 |       2212 |   1 |  -1 |   5 |   5 |   -0.053 |   -0.064 |   -0.035 |    0.915 | **0.938 | M = 0.910
|   3 |          Cl39 |   2 | 1000170390 |   1 |  -1 |  11 |  11 |    0.053 |    0.064 |    0.035 |   36.301 | *36.290 | M = 36.301
|   4 |           mu- |   1 |         13 |   0 |  -1 |  -1 |  -1 |    0.077 |   -0.292 |    1.610 |    1.641 |   0.106 | P = (-0.047,0.178,-0.983)
|   5 |       Delta++ |   3 |       2224 |   2 |  -1 |   6 |   7 |   -0.130 |    0.228 |    0.385 |    1.303 | **1.231 | M = 1.216
|   6 |        proton |  14 |       2212 |   5 |  -1 |   8 |   8 |   -0.150 |    0.392 |    0.324 |    1.078 |   0.938 | FSI = 1
|   7 |           pi+ |  14 |        211 |   5 |  -1 |   9 |  10 |    0.020 |   -0.164 |    0.061 |    0.225 |   0.140 | FSI = 4
|   8 |        proton |   1 |       2212 |   6 |  -1 |  -1 |  -1 |   -0.150 |    0.392 |    0.324 |    1.078 |   0.938 |
|   9 |        proton |   1 |       2212 |   7 |  -1 |  -1 |  -1 |    0.104 |    0.476 |   -0.152 |    1.068 |   0.938 |
|  10 |        proton |   1 |       2212 |   7 |  -1 |  -1 |  -1 |   -0.070 |   -0.429 |    0.056 |    1.035 |   0.938 |
|  11 |      HadrBlob |  15 | 2000000002 |   3 |  -1 |  -1 |  -1 |    0.039 |   -0.148 |    0.192 |   34.422 | **0.000 | M = 34.421
|-----------------------------------------------------------------------------------------------------------------------|
|       Fin-Init:                                                 |    0.080 |    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] : 0000000000000000    |  1st set:                                                       none |
| Err mask [bits:15->0] : 1111111111111111    |  Is unphysical:    NO |   Accepted:   YES                            |
|-----------------------------------------------------------------------------------------------------------------------|
| sig(Ev) =       1.17152e-37 cm^2  | d2sig(W,Q2;E)/dWdQ2 =     1.11155e-36 cm^2/GeV^3 | Weight =          1.00000 |
|-----------------------------------------------------------------------------------------------------------------------|
```

```
-------------------------------------------------------------------------------------------------------------
GENIE Interaction Summary
-------------------------------------------------------------------------------------------------------------
[-] [Init-State]
 |--> probe        : PDG-code = 14 (nu_mu)
 |--> nucl. target : Z = 18, A = 40, PDG-Code = 1000180400 (Ar40)
 |--> hit nucleon  : PDC-Code = 2212 (proton)
 |--> hit quark    : no set
 |--> probe 4P     : (E =     2.029360, Px =     0.000000, Py =     0.000000, Pz =     2.029360)
 |--> target 4P    : (E =    37.215526, Px =     0.000000, Py =     0.000000, Pz =     0.000000)
 |--> nucleon 4P   : (E =     0.914517, Px =    -0.053184, Py =    -0.064262, Pz =    -0.034694)
[-] [Process-Info]
 |--> Interaction : Weak[CC]
 |--> Scattering  : RES
[-] [Kinematics]
 |--> *Selected* Bjorken x = 0.151851
 |--> *Selected* Inelasticity y = 0.199377
 |--> *Selected* Momentum transfer Q2 (>0) = 0.116639
 |--> *Selected* Hadronic invariant mass W = 1.216411
[-] [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
```

- HEP format events for the contained particles
  - The particles are a bit enriched in GENIE to add polarisation information
    - Although the information is not correctly used yet
  - GENIE massively extended on the status of the particle
    - src/Framework/GHEP/GHepStatus.h
- Summary is information used in the creation of the event
  - Its definition is here
  - But it's useful to categorise the event
  - It also contain some kinematic variables
    - Keep in mind that they are model dependent
    - So be careful, if you need this information, extract it yourself
- you can get these displays from
  - gevdump
  - the status file when you generate events

Configuration

- All the configurations live in Generator/config
  - Some configuration will live in Reweight/config but for now it's all empty
  - Only exception is the PDG data
    - https://github.com/GENIE-MC/Generator/tree/master/data/evgen/catalogues/pdg
- Each GENIE algorithm has an xml file to configure it
  - The system looks for them in $GENIE/config
  - The mapping between the name of the algorithm and its xml file is also an xml file
    - https://github.com/GENIE-MC/Generator/blob/master/config/master_config.xml
- Files that are the same for every tune are in the top directory
  - The tune directories override the xml files in the top directory
  - From the technical point of view this is what a tune is
    - A collection of files that supersedes the top config directory
    - The directory has to have a particular name structure
- using the option --xml-path you can even set a path to directory
  - The files contained in that directory takes precedence on every other location

Configuration details

- A parameter will be looked in the following (in order or priority)
  - The xml file associated to the algorithm that looks for the parameter
  - A block in the CommonParam.xml file with the name linked byt the xml file of the algorithm
    - e.g. <param type="string" name="CommonParam"> MultiNucleons </param> links the block MultiNucleons from CommonParam.xml
  - parameters will also be looked in the xml files for Subalgo
    - A subalgo is another algorithm linked in the xml file
      - e.g. <param type="alg" name="AxialFormFactorModel"> genie::DipoleAxialFormFactorModel/Default </param>

- Notable xml files
  - CommonParam.xml
  - TuneGeneratorList.xml -> Sets the algorithm chain for each generator chain
  - EventGeneratorListAssembler.xml -> sets the list of Event generators corresponding to a set
    - See option --event-generator
  - ModelConfiguration.xml -> sets which model to use for each event generator chain
    - Defined in tune subdir
  - TuneGeneratorList.xml -> sets the default EventGeneratorList for a tune
    - Defined in tune subdir

Exercises

Exercise 0 - Installation

- I’m going to assume you have a working GENIE version
- But if not,
  - I’m here to see what we can do
- This is the perfect place to thank the tutors that helped you last week:
  - Thank you all as without you it would have been impossible to run this exercise
    - Valentina Cicero
    - Filippo Mei
    - Valerio Pia
    - Francesco Poppi
    - Elisa Sanzani

Exercise 1 - Event generation 1 - Spline generation

- Create your own splines
  - Can you control which splines do you put in the splines?
    - Neutrinos flavour, target, processes, number of points
- Give it a try, and see what you obtain
  - If it's a full set, probably too long, stop the process
  - But before you stop it, have a look at the text and understand the output
- Can you get the splines from the genie website
  - Can you locate the actual files in the tarball?
- Can you create a ROOT file of the splines using gspl2root?
  - Can you make plots of splines? Do they make sense to you?

Exercise 2 - Event generation 2 - Actual event generation

- Generate events using the splines you have via gevgen
  - Start with a small sample - around 100 events
  - Which files are generated?
    - Can you control them? Change name for example
  - Experiment a bit with the options
    - Flux
    - Target composition
- Use native tools to navigate the files
  - gevdump and gntpc
    - Use gevdump to have a look at the events
    - Can you understand EVERY detail of the event?
  - Use gntpc to obtain a gst tree
    - Can you make simple plots out of one of the variable?
      - Does it match your expectations?
    - quickly check all the variables to see if they make sense to you

Exercise 3 - Event navigation and analyses

- An easy way to navigate the events you generated is to read the gst tree with ROOT
  - But this does not contain all the information
- In GENIE there are a number of simple scripts you can use to explore the event you created
  - e.g. https://github.com/GENIE-MC/Generator/blob/master/src/contrib/delta_decay/plots.C
- Write your own script and run it via the genie root wrapper
  - $ genie
  - $ .L my_script.C+
  - $ my_function()
- Try to extract variables that are not available. Here is a list as an example in increasing level of difficulty:
  - Mandelstam s
  - Some TKI variables
  - Integrated cross section of a pion in the nucleus
- Can you generate distributions of any variable, and normalise them the cross section?
  - Refinement: the goal is to obtain exclusive cross section distribution

Exercise 4 - Create configurations

- Locate the xml file that contains a particular variable that affect your event generation
  - E.g. QEL axial mass, SRC-fraction
- Copy the file you plan to change in another directory
  - Can you run forcing GENIE to use that xml file you created and changed?
  - Remove the parameter: does GENIE crashes? Can you understand the error?
    - Or does it not crash? Why?
- Can you create a new tune?
  - And does it work when you run it?

Backup