Monte-Carlo generators in ALICE

Jochen Klein (CERN)

December 17, 2015
focus on **generator level** productions using generators that produce tracks

- feed into **Rivet** for analysis
- feed into **AliRoot** for ALICE analysis task requires interfacing

- **Caveat:** still evolving!

- look at
  - [http://jklein.web.cern.ch/jklein/mc](http://jklein.web.cern.ch/jklein/mc)
modes of operation

- **generator level**
  - Rivet
    - direct comparison of generators
    - used by MC community
  - AliRoot
    - usage of ALICE analysis framework
      (convenient when working on data analysis)

- **full simulation**
  - includes detector effects
  - detector simulation is rather decoupled
    (not further discussed here)

- **fast simulation**
  - use parameterized detector behaviour
  - under development
    (deserves a talk of its own)
Rivet introduction

Rivet

- reads HepMC input
- runs one or more analyses
- writes histograms
- produces (comparison) plots
producing input to Rivet

- use your favourite generator to produce HepMC data
  here we use DIPSY because it’s of interest recently and works everywhere

- first you need a (generator-specific) configuration file,
  e.g. copy the one from

  ```
  ${ALICE_PHYSICS}/PWG/HMTF/MonteCarlo/DIPSYpp_HepMC.in
  ```

  (it’s not coupled to AliPhysics though)

- e.g. for DIPSY on lxplus:

  ```
  alienv enter ThePEG::v2015-08-11-4
  setupThePEG \n    -r ${THEPEG_ROOT}/lib/ThePEG/ThePEGDefaults.rpo \n    -I ${THEPEG_ROOT}/share/Ariadne \n    DIPSYpp_HepMC.in
  runThePEG DIPSYpp.run -N 10
  ```
Running a Rivet analysis

- run a Rivet analysis on the HepMC output
- e.g. on lxplus

```
alienv enter Rivet::2.2.1-2
export PYTHONPATH=${RIVET_ROOT}/lib/python2.7/site-packages/:$PYTHONPATH
rivet -a ALICE_2010_S8625980 DIPSYpp.hepmc
rivet-mkhtml ...
```

- look at the plots in plots/
- now you can try with
  - another analysis
  - more statistics
  - different configuration
  - ...

Rivet without intermediate file

- you can avoid the intermediate HepMC file by using a FIFO instead:

  mkfifo /tmp/fifo_$UID.hepmc

  (FIFO not supported on afs)

- start the generator (writing to that file)

- start Rivet (reading from that file)

- nothing ever gets written to disk

- try it ...
AliRoot uses dedicated structures to store MC information: *AliMCEvent*, *AliStack*, *AliMCParticle*, ...  

- need interface  
  to translate generator output to ALICE structures  
- AliRoot uses generators inheriting from AliGenMC to  
  - steer the generation process  
  - produce Monte-Carlo particles  
readers

- translation from generator format to AliRoot, either in AliGenMC derivative or dedicated reader
- for generators with HepMC output, use common reader:
  - generator
  - HepMC
  - AliGenReaderHepMC
  - Kinematics.root
- for other generators dedicated implementations needed, e.g.
  - AliGenEposReader (for EPOS 3)
  - ...
external executables (AliGenExtExec)

- recently support was added to run external generators, behind the scenes:
  - prepare files for data exchange
  - spawn subjob for generator
  - read next event when requested
- generator process controlled by bash script allows arbitrary generator setup
on-the-fly LEGO trains

- datasets used to define generator and settings
  - need Add-macro for one of the generators listed before, (AliGenExtExec for external generators)

- analysis wagons setup as for normal data analysis

- train runs per dataset, i.e. generator,
  (at the moment upon request)

- particularly attractive when generation is fast
  (very flexible, no I/O overhead, ...)

- for slow generation storage of the events better
example: DIPSY in LEGO train

- look at MCGen_pp train
- dataset Test_DIPSY uses
  
  \texttt{ANALYSIS/macros/train/AddMCGenExtExec.C}

  to add generator to steer external process

- actual generator and options chosen by configuration:

  \begin{verbatim}
  ((AliGenExtExec*) generator)->
  SetPathScript("$ALICE_PHYSICS/PWG/MCLEGO/ThePEG/gen_dipsy_ropes.sh";
  \end{verbatim}
dedicated productions (runmc)

- run generator and write Kinematics.root files to alien
- analysis can be run on these files
- tools developed for automatization,
  e.g. on lxplus:

  alienv enter AliPhysics::vAN-20151216-1,ThePEG::v2015-08-11-4
  ${ALICE_PHYSICS}/PWG/HMTF/MonteCarlo/runmc -h

  gives you help and currently used options
- local production

  ${ALICE_PHYSICS}/PWG/HMTF/MonteCarlo/runmc -L

  producing Kinematics.root locally
- can serve as input for standard analysis
grid production

- use runmc to prepare and copy files for grid production:
  alien-token-init
  ${ALICE_PHYSICS}/PWG/HMTF/MonteCarlo/runmc -G

- produces JDL to submit grid jobs
  ready for submission
down-scaling

- reject events to produce a sample which is (more) homogeneous in some observable
- used to flatten the multiplicity distribution (HMTF), need down-scaling factors for given multiplicity
- examples contained in runmc
- macros to extract multiplicity down-scaling factors exist, but not integrated yet
available generators

<table>
<thead>
<tr>
<th>generator (package name)</th>
<th>distributed via</th>
<th>versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMPT</td>
<td>AliRoot</td>
<td>v1.25t3/v2.25t3</td>
</tr>
<tr>
<td>DIPSY (ThePEG)</td>
<td>AliRoot</td>
<td>v2015-08-11 (private comm.)</td>
</tr>
<tr>
<td>DPMJET</td>
<td>AliRoot</td>
<td>3.0.5</td>
</tr>
<tr>
<td>EPOS</td>
<td>package</td>
<td>3.111</td>
</tr>
<tr>
<td>EPOS LHC (CRMC)</td>
<td>package</td>
<td>1.5.4</td>
</tr>
<tr>
<td>Herwig</td>
<td>AliRoot</td>
<td>6.507, 6.510</td>
</tr>
<tr>
<td>HIJING</td>
<td>AliRoot</td>
<td>1.35, 1.36</td>
</tr>
<tr>
<td>JEWEL</td>
<td>package</td>
<td>2.0.2</td>
</tr>
<tr>
<td>Pythia 6</td>
<td>AliRoot</td>
<td>6.4.21, 6.4.24, 6.4.28</td>
</tr>
<tr>
<td>Pythia 8</td>
<td>AliRoot/package</td>
<td>8.175, 8.205, 8.210</td>
</tr>
<tr>
<td>Starlight</td>
<td>AliRoot</td>
<td>r193</td>
</tr>
</tbody>
</table>

Disclaimers:

- to be extended
- not all of them tested personally now
further plans

- extend documentation
- add further generators (Herwig++, Sherpa, ...)
- preparation of common generator setup for
  - Rivet
  - LEGO trains
  - full simulation
Summary

- several options to run Monte-Carlo generators which is best depends on the purpose
- work on-going to make further generators usable on LEGO trains and grid
- thanks to people involved in these efforts
- feedback and ideas are very welcome!
References

- twiki: HMTF generator-level production
- this documents as web page, pdf, slides
- individual generator documentation