HIJING++ and HEPMC

The HEPMC standards in heavy ion Monte Carlo models

GÁBOR BÍRÓ

07 06 2021

Wigner Research Centre for Physics

HIJING++ and HEPMC

(Final version is not published yet)

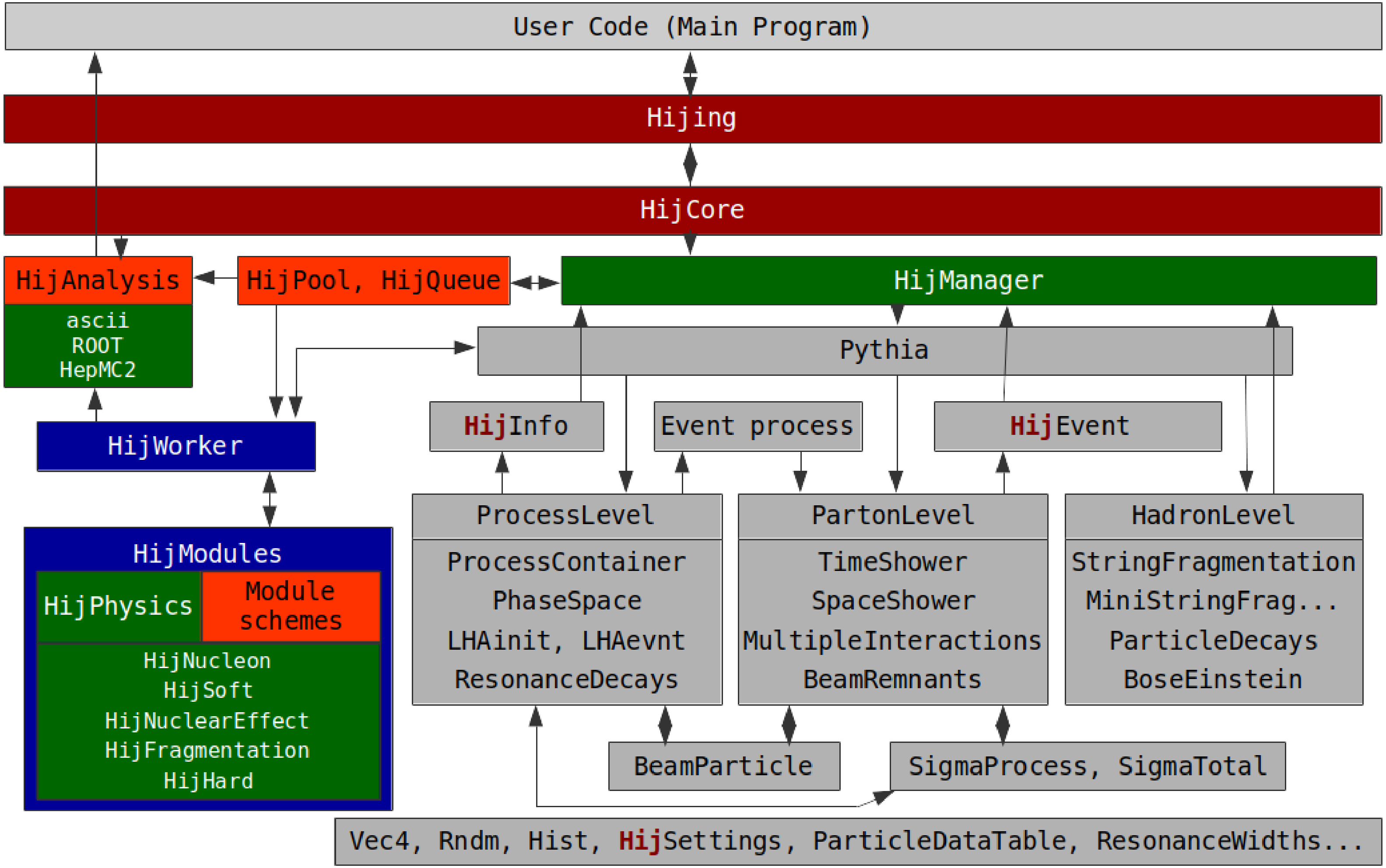
The HEPMC standards in heavy ion Monte Carlo models

GÁBOR BÍRÓ

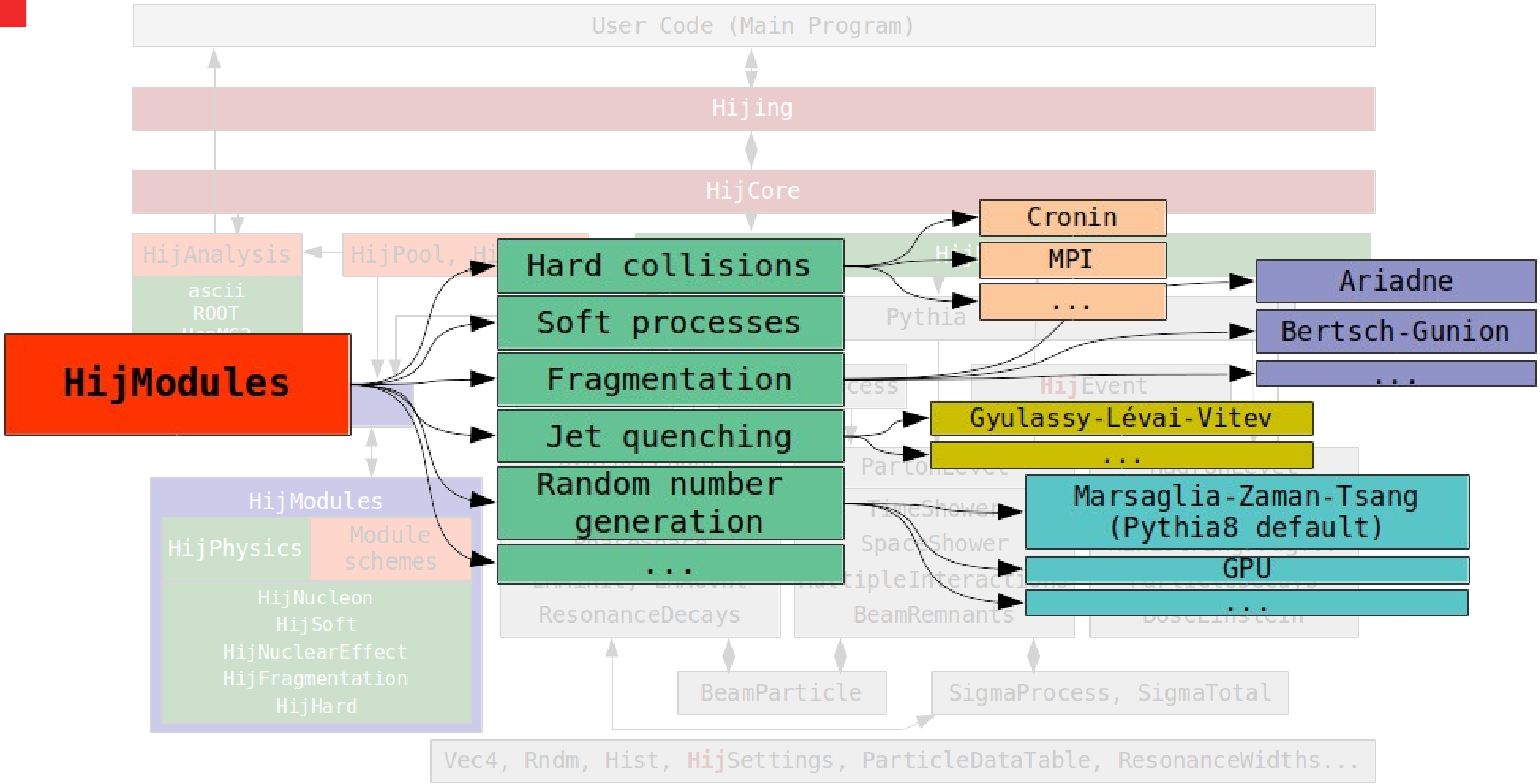
07 06 2021

Wigner Research Centre for Physics

HIJING++ structure



HIJING++ structure



User Code (Main Program)

Hijing

HijCore

HijAnalysis

ascii

ROOT

HijPool, Hi

Hij

Pythia

HijEvent

PartonLevel

HadronLevel

TimeShower

SpaceShower

MultipleInteractions

BeamRemnants

ResonanceDecays

ParticleDecays

BoseEinstein

BeamParticle

SigmaProcess, SigmaTotal

Vec4, Rndm, Hist, HijSettings, ParticleDataTable, ResonanceWidths...

HijModules

HijModules

HijPhysics

Module schemes

HijNucleon

HijSoft

HijNuclearEffect

HijFragmentation

HijHard

Hard collisions

Soft processes

Fragmentation

Jet quenching

Random number generation

...

Cronin

MPI

...

Ariadne

Bertsch-Gunion

...

Gyulassy-Lévai-Vitev

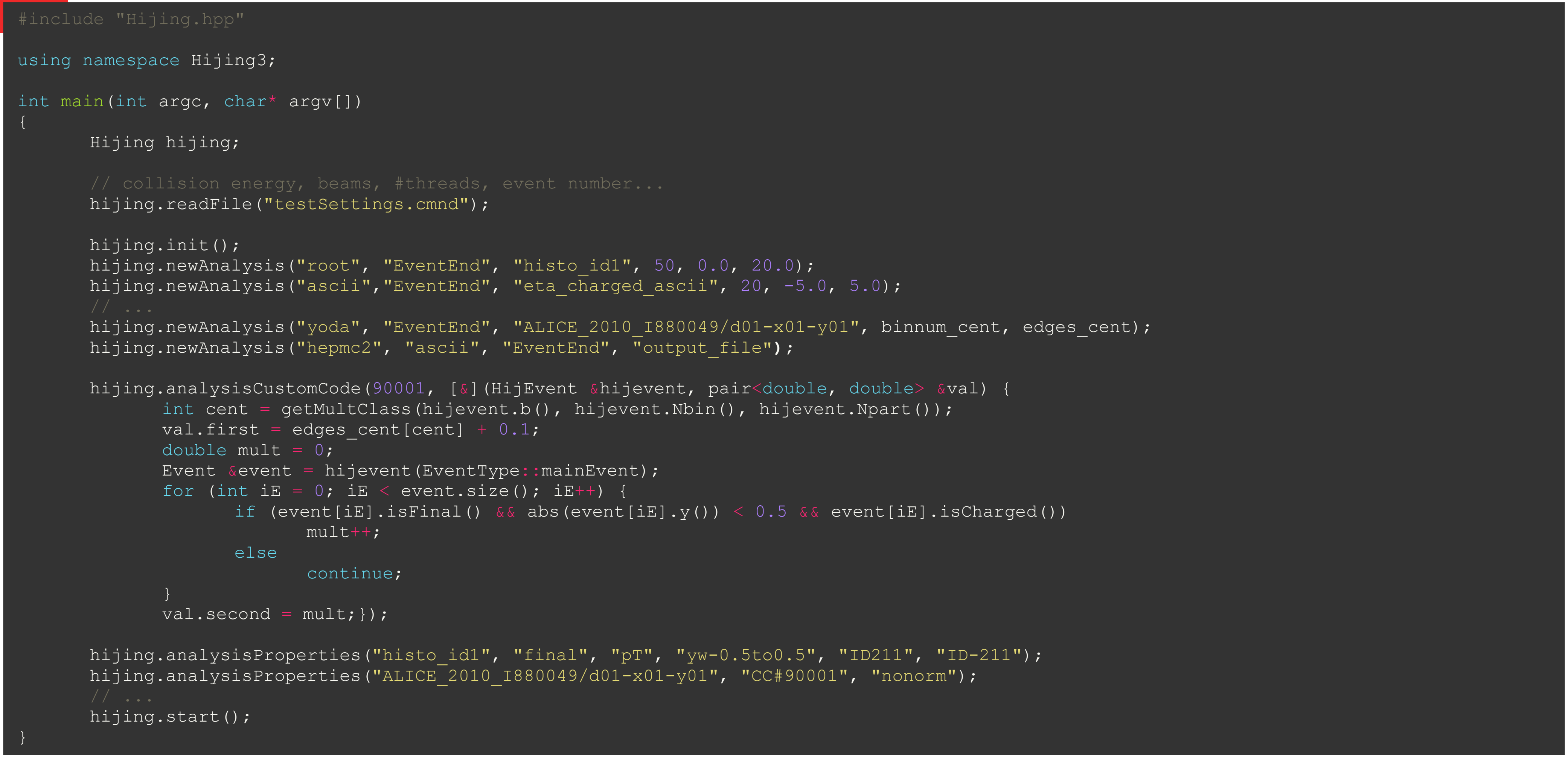
...

Marsaglia-Zaman-Tsang (Pythia8 default)

GPU

...

HIJING++ analysis definitions



```
#include "Hijing.hpp"

using namespace Hijing3;

int main(int argc, char* argv[])
{
        Hijing hijing;

        // collision energy, beams, #threads, event number...
        hijing.readFile("testSettings.cmnd");

        hijing.init();
        hijing.newAnalysis("root", "EventEnd", "histo_id1", 50, 0.0, 20.0);
        hijing.newAnalysis("ascii","EventEnd", "eta_charged_ascii", 20, -5.0, 5.0);
        // ...
        hijing.newAnalysis("yoda", "EventEnd", "ALICE_2010_I880049/d01-x01-y01", binnum_cent, edges_cent);
        hijing.newAnalysis("hepmc2", "ascii", "EventEnd", "output_file");

        hijing.analysisCustomCode(90001, [&](HijEvent &hijevent, pair<double, double> &val) {
                int cent = getMultClass(hijevent.b(), hijevent.Nbin(), hijevent.Npart());
                val.first = edges_cent[cent] + 0.1;
                double mult = 0;
                Event &event = hijevent(EventType::mainEvent);
                for (int iE = 0; iE < event.size(); iE++) {
                        if (event[iE].isFinal() && abs(event[iE].y()) < 0.5 && event[iE].isCharged())
                                mult++;
                        else
                                continue;
                }
                val.second = mult;});

        hijing.analysisProperties("histo_id1", "final", "pT", "yw-0.5to0.5", "ID211", "ID-211");
        hijing.analysisProperties("ALICE_2010_I880049/d01-x01-y01", "CC#90001", "nonorm");
        // ...
        hijing.start();
}
```

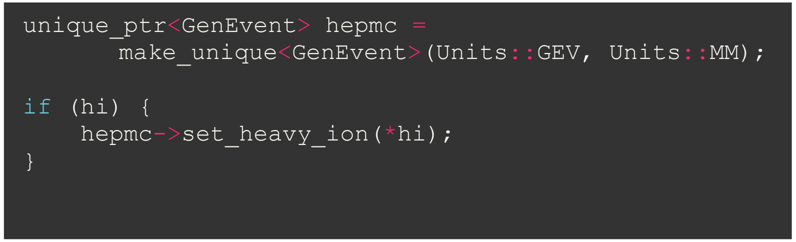
HIJING++ tuning with Rivet

Rivet 3.x:

Currently only HepMC2 is supported → the migration shouldn't be very cumbersome (http://hepmc.web.cern.ch/hepmc/differences.html)

In pp mode: works well

In HI mode: ***should*** work well...

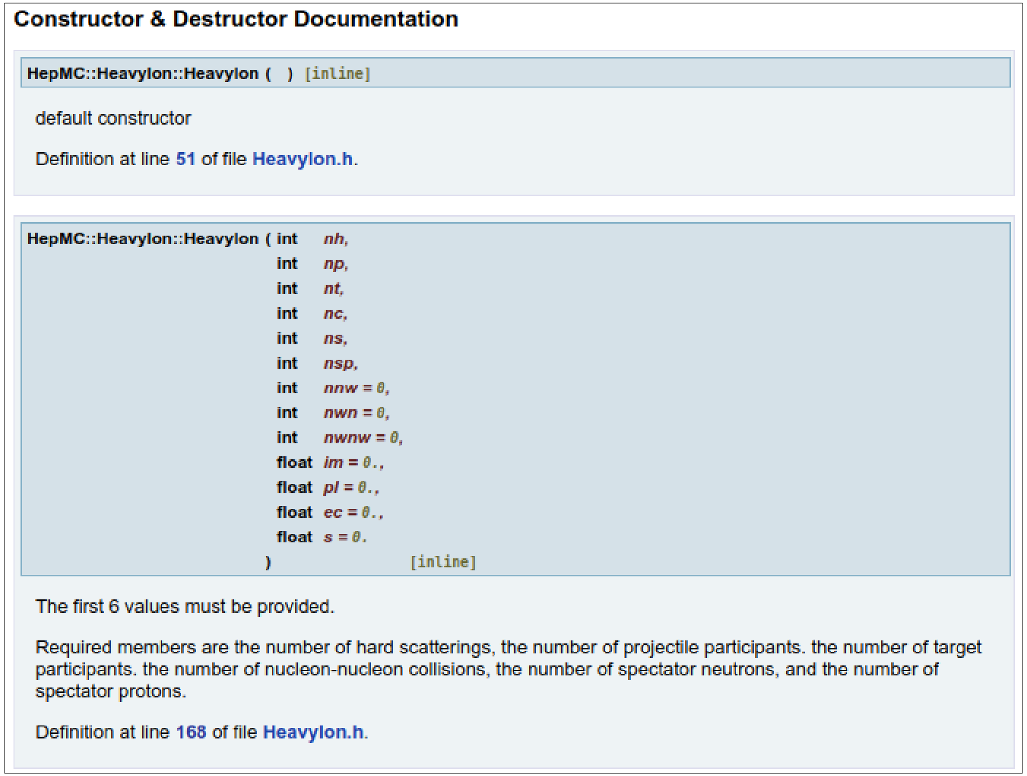


```
unique_ptr<GenEvent> hepmc =
      make_unique<GenEvent>(Units::GEV, Units::MM);

if (hi) {
    hepmc->set_heavy_ion(*hi);
}
```

Where the heavy ion object is constructed as:

→ each argument is provided



Constructor & Destructor Documentation

```
HepMC::HeavyIon::HeavyIon (  ) [inline]
```

default constructor

Definition at line 51 of file HeavyIon.h.

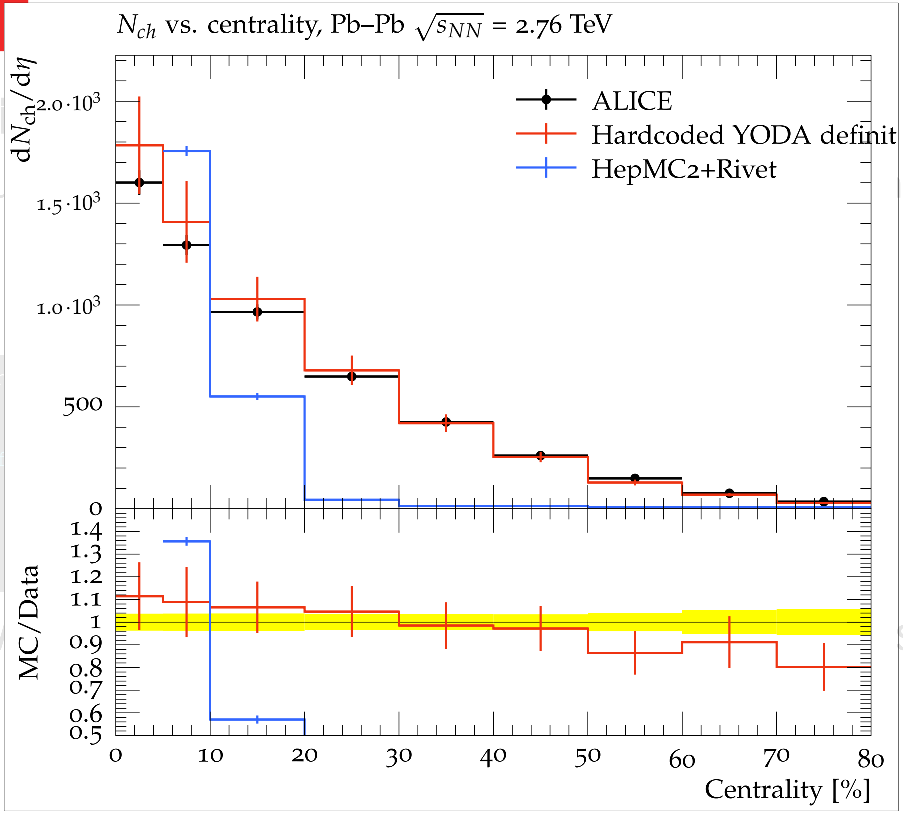
```
HepMC::HeavyIon::HeavyIon ( int   nh,
                            int   np,
                            int   nt,
                            int   nc,
                            int   ns,
                            int   nsp,
                            int   nnw = 0,
                            int   nwn = 0,
                            int   nwnw = 0,
                            float im = 0.,
                            float pl = 0.,
                            float ec = 0.,
                            float s = 0.
                          )              [inline]
```

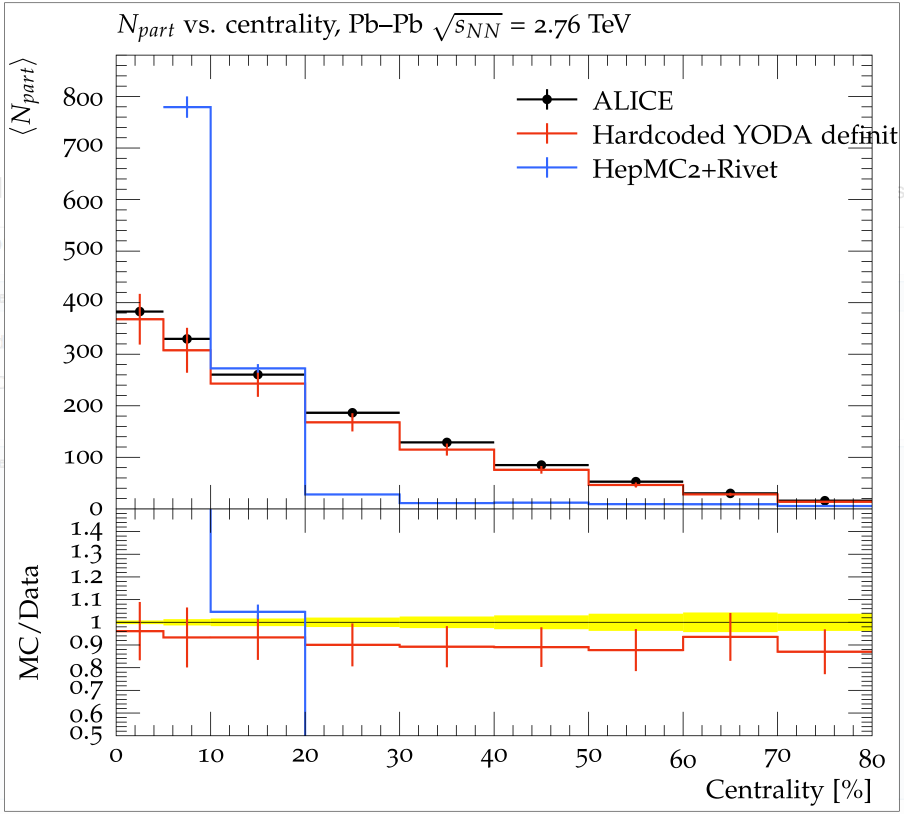
The first 6 values must be provided.

Required members are the number of hard scatterings, the number of projectile participants. the number of target participants. the number of nucleon-nucleon collisions, the number of spectator neutrons, and the number of spectator protons.

Definition at line 168 of file HeavyIon.h.

HIJING++ tuning with Rivet





```
Rivet.Analysis.ALICE_2010_I880049: INFO  Found calibration histogram REF /REF/ALICE_2015_PBPBCentrality/V0M
Rivet.Projection.SingleValueProjection: INFO  Constructing PercentileProjection from /REF/ALICE_2015_PBPBCentrality/V0M
```

Something is clearly wrong somewhere…

Issues, questions

Particle decays in HIJING++

Handled with Pythia8

Technically challenging at events with large particle number, but doable

Impact parameter for centrality calibration

Using HepMC output with Rivet

With FIFO it is straightforward → only single threaded mode!

Usage of multiple FIFO for multithreaded mode?

W/O FIFO, file sizes grow very rapidly

(Derived quantities with HepMC+Rivet? E.g. nuclear modification factor)

Maybe just lack of knowledge → any guide is welcome