Implementation of EventEvaluator output in sidis-eic

SIDIS WG meeting
June 28, 2022

Ralf Seidl (RIKEN)

EventEvaluator output

```
 TTree          *fChain;   //!pointer to the analyzed TTree or TChain
   Int_t           fCurrent; //!current Tree number in a TChain
// Fixed size dimensions of array or collections stored in the TTree if any.
   // Declaration of leaf types
   Int_t           nTracks;
   Float_t         tracks_ID[40];   //[nTracks]
   Float_t         tracks_px[40];   //[nTracks]
   Float_t         tracks_py[40];   //[nTracks]
   Float_t         tracks_pz[40];   //[nTracks]
   Float_t         tracks_trueID[40];   //[nTracks]
Int_t           nMCPart;
   Float_t         mcpart_ID[7000];   //[nMCPart]
   Float_t         mcpart_ID_parent[7000];   //[nMCPart]
   Float_t         mcpart_PDG[7000];   //[nMCPart]
   Float_t         mcpart_E[7000];   //[nMCPart]
   Float_t         mcpart_px[7000];   //[nMCPart]
   Float_t         mcpart_py[7000];   //[nMCPart]
   Float_t         mcpart_pz[7000];   //[nMCPart]
 Int_t           nHepmcp;
   Int_t           hepmcp_procid;
   Float_t         hepmcp_x1;
   Float_t         hepmcp_x2;
   Float_t         hepmcp_ID[150];   //[nHepmcp]
   Float_t         hepmcp_ID_parent[150];   //[nHepmcp]
   Float_t         hepmcp_status[150];   //[nHepmcp]
   Float_t         hepmcp_PDG[150];   //[nHepmcp]
   Float_t         hepmcp_E[150];   //[nHepmcp]
   Float_t         hepmcp_px[150];   //[nHepmcp]
   Float_t         hepmcp_py[150];   //[nHepmcp]
   Float_t         hepmcp_pz[150];   //[nHepmcp]
   Int_t           hepmcp_m1[150];   //[nHepmcp]
   Int_t           hepmcp_m2[150];   //[nHepmcp]
```

SvtxTrackMap node

- Relates to trueID from PHG4TruthInfoContainer

PHG4TruthInfoContainer node → PHG4Particle:

- Only final state/secondary particles
- Neg ID: secondary
- ancestry

HEPMC output (not yet in official EE):

- Process ID (Pythia 99, 131/132, 135/136, etc)
- All generated particles: use status (4/3/0/1) for initial/PS/decaying/final info (not Pythia!!!)
- Parton info

Various other evaluators not shown here (Calorimeters)

RIKEN

AnalysisEE added

- Very closely following the AnalysisDD4hep class but using EventEvaluator output for tracks, HEPMC and G4Truth parts so far

main ▾ sidis-eic / src / Go to file

This branch is 1 commit ahead of c-dilks:main. Contribute ▾

Ralf Seidl Added AnalysisEE class to read in Fun4All related EventEvaluator gene... ... 97aff5a 22 minutes ago History

..		
interp	Initial commit	9 months ago
sfset	Use updated grids for Sivers asymmetry to lower x	8 months ago
Analysis.cxx	resolve: hadronic recon methods not working in fullsim (c-dilks#122)	7 months ago
Analysis.h	rename largex-eic to sidis-eic (c-dilks#138)	last month
AnalysisDD4hep.cxx	resolve: hadronic recon methods not working in fullsim (c-dilks#122)	7 months ago
AnalysisDD4hep.h	resolve: hadronic recon methods not working in fullsim (c-dilks#122)	7 months ago
AnalysisDelphes.cxx	automate delphes for S3 hepmc files (c-dilks#133)	7 months ago
AnalysisDelphes.h	rename largex-eic to sidis-eic (c-dilks#138)	last month
AnalysisEE.cxx	Added AnalysisEE class to read in Fun4All related EventEvaluator gene...	22 minutes ago
AnalysisEE.h	Added AnalysisEE class to read in Fun4All related EventEvaluator gene...	22 minutes ago
BinSet.cxx	add GetMin and GetMax to BinSet	8 months ago
BinSet.h	rename largex-eic to sidis-eic (c-dilks#138)	last month

Todo

- So far, no clusters implemented, nor track projections to actually use clusters (needed for vetoing tracks, or using better EMCAL electron information)
- Possibility of process selection (LO, PGF, QCDC, diffractive, etc – only related to Pythia6 output anyway)
- Using hadron or electron(?) Likelihood ratios for PID information (so far use truth info)
- Check that crossing angle is properly addressed
- Cross check consistency of kinematics with strandalone code
- Issues: Even in Truth particles electron energies sometimes similar or even slightly smaller as momentum (precision issue?) → use PDG mass for now, need to check EventEvaluator

Summary

- First contribution of EventEvaluator output to sidis-eic added (PR #148) in https://github.com/c-dilks/sidis-eic
- Tested some EE output files using simple Tree (file list documentation outdated), not tested histo output, etc

Initialization (create tree, add tracks to it)

```
int EventEvaluatorEIC::Init(PHCompositeNode* topNode)
{
  _ievent = 0;

  _tfile = new TFile(_filename.c_str(), "RECREATE");

  _event_tree = new TTree("event_tree", "event_tree");
```

```
if (_do_TRACKS)
{
  _event_tree->Branch("nTracks", &_nTracks, "nTracks/I");
  _event_tree->Branch("tracks_ID", _track_ID, "tracks_ID[nTracks]/F");
  _event_tree->Branch("tracks_charge", _track_charge, "tracks_charge[nTracks]/S");
  _event_tree->Branch("tracks_px", _track_px, "tracks_px[nTracks]/F");
  _event_tree->Branch("tracks_py", _track_py, "tracks_py[nTracks]/F");
  _event_tree->Branch("tracks_pz", _track_pz, "tracks_pz[nTracks]/F");
  _event_tree->Branch("tracks_x", _track_x, "tracks_x[nTracks]/F");
  _event_tree->Branch("tracks_y", _track_y, "tracks_y[nTracks]/F");
  _event_tree->Branch("tracks_z", _track_z, "tracks_z[nTracks]/F");
  _event_tree->Branch("tracks_ndf", _track_ndf, "tracks_ndf[nTracks]/F");
  _event_tree->Branch("tracks_chi2", _track_chi2, "tracks_chi2[nTracks]/F");
  _event_tree->Branch("tracks_dca", _track_dca, "tracks_dca[nTracks]/F");
  _event_tree->Branch("tracks_dca_2d", _track_dca_2d, "tracks_dca_2d[nTracks]/F");
  _event_tree->Branch("tracks_trueID", _track_trueID, "tracks_trueID[nTracks]/F");
  _event_tree->Branch("tracks_source", _track_source, "tracks_source[nTracks]/s");

  if (_do_PID_LogLikelihood)
```

G4Truthparticles (includes secondaries, etc)

```
if (_do_MCPARTICLES)
{
  PHG4TruthInfoContainer* truthinfocontainer = findNode::getClass<PHG4TruthInfoContainer>(topNode, "G4TruthInfo");
  if (truthinfocontainer)
  {
    if (Verbosity() > 0)
    {
      cout << "saving MC particles" << endl;
    }
    //GetParticleRange for all particles
    //GetPrimaryParticleRange for primary particles
    PHG4TruthInfoContainer::ConstRange range = truthinfocontainer->GetParticleRange();
    for (PHG4TruthInfoContainer::ConstIterator truth_itr = range.first; truth_itr != range.second; ++truth_itr)
    {
      PHG4Particle* g4particle = truth_itr->second;
      if (!g4particle) continue;

      int mcSteps = 0;
      PHG4Particle* g4particleMother = truth_itr->second;
```

```
        _mcpart_ID[_nMCPart] = g4particle->get_track_id();
        _mcpart_ID_parent[_nMCPart] = g4particle->get_parent_id();
        _mcpart_PDG[_nMCPart] = g4particle->get_pid();
        _mcpart_E[_nMCPart] = g4particle->get_e();
        _mcpart_px[_nMCPart] = g4particle->get_px();
        _mcpart_py[_nMCPart] = g4particle->get_py();
        _mcpart_pz[_nMCPart] = g4particle->get_pz();
        PHG4VtxPoint* vtxtmp = truthinfocontainer->GetVtx(g4particle->get_vtx_id());
        if (vtxtmp)
        {
          _mcpart_x[_nMCPart] = vtxtmp->get_x();
          _mcpart_y[_nMCPart] = vtxtmp->get_y();
          _mcpart_z[_nMCPart] = vtxtmp->get_z();
        }
        //BCID added for G4Particle --  HEPMC particle matching
        _mcpart_BCID[_nMCPart] = g4particle->get_barcode();
        // TVector3 projvec(_mcpart_px[0],_mcpart_py[0],_mcpart_pz[0]);
        // float projeta = projvec.Eta();
        _nMCPart++;
      }
```


RIKEN

HEPMC information (generator info)

```
_nHepmcp = 0;
if (_do_HEPMC)
{
  PHHepMCGenEventMap* hepmceventmap = findNode::getClass<PHHepMCGenEventMap>(topNode, "PHHepMCGenEventMap");
  if (hepmceventmap)
  {
    if (Verbosity() > 0)
    {
      cout << "saving HepMC output" << endl;
    }
    if (Verbosity() > 0)
    {
      hepmceventmap->Print();
    }

    for (PHHepMCGenEventMap::ConstIter eventIter = hepmceventmap->begin();
         eventIter != hepmceventmap->end();
         ++eventIter)
    {
      PHHepMCGenEvent* hepmcevent = eventIter->second;

        _hepmcp_x1 = pdfinfo->x1();
        _hepmcp_x2 = pdfinfo->x2();
        _hepmcp_Q2 = pdfinfo->scalePDF();

        // m_mpi = truthevent->mpi();

        _hepmcp_procid = truthevent->signal_process_id();

        if (Verbosity() > 2)
        {
          cout << " Iterating over an event" << endl;
        }
        for (HepMC::GenEvent::particle_const_iterator iter = truthevent->particles_begin();
             iter != truthevent->particles_end();
             ++iter)
        {
          _hepmcp_E[_nHepmcp] = (*iter)->momentum().e();
          _hepmcp_PDG[_nHepmcp] = (*iter)->pdg_id();
          _hepmcp_px[_nHepmcp] = (*iter)->momentum().px();
          _hepmcp_py[_nHepmcp] = (*iter)->momentum().py();
          _hepmcp_pz[_nHepmcp] = (*iter)->momentum().pz();
          _hepmcp_status[_nHepmcp] = (*iter)->status();
          _hepmcp_BCID[_nHepmcp] = (*iter)->barcode();
          _hepmcp_m2[_nHepmcp] = 0;
          _hepmcp_m1[_nHepmcp] = 0;
          if ((*iter)->production_vertex())
          {
            for (HepMC::GenVertex::particle_iterator mother = (*iter)->production_vertex()->particles_begin(HepMC::parents);
                 mother != (*iter)->production_vertex()->particles_end(HepMC::parents);
                 ++mother)
            {
              _hepmcp_m2[_nHepmcp] = (*mother)->barcode();
              if (_hepmcp_m1[_nHepmcp] == 0)
                _hepmcp_m1[_nHepmcp] = (*mother)->barcode();
            }
          }
```