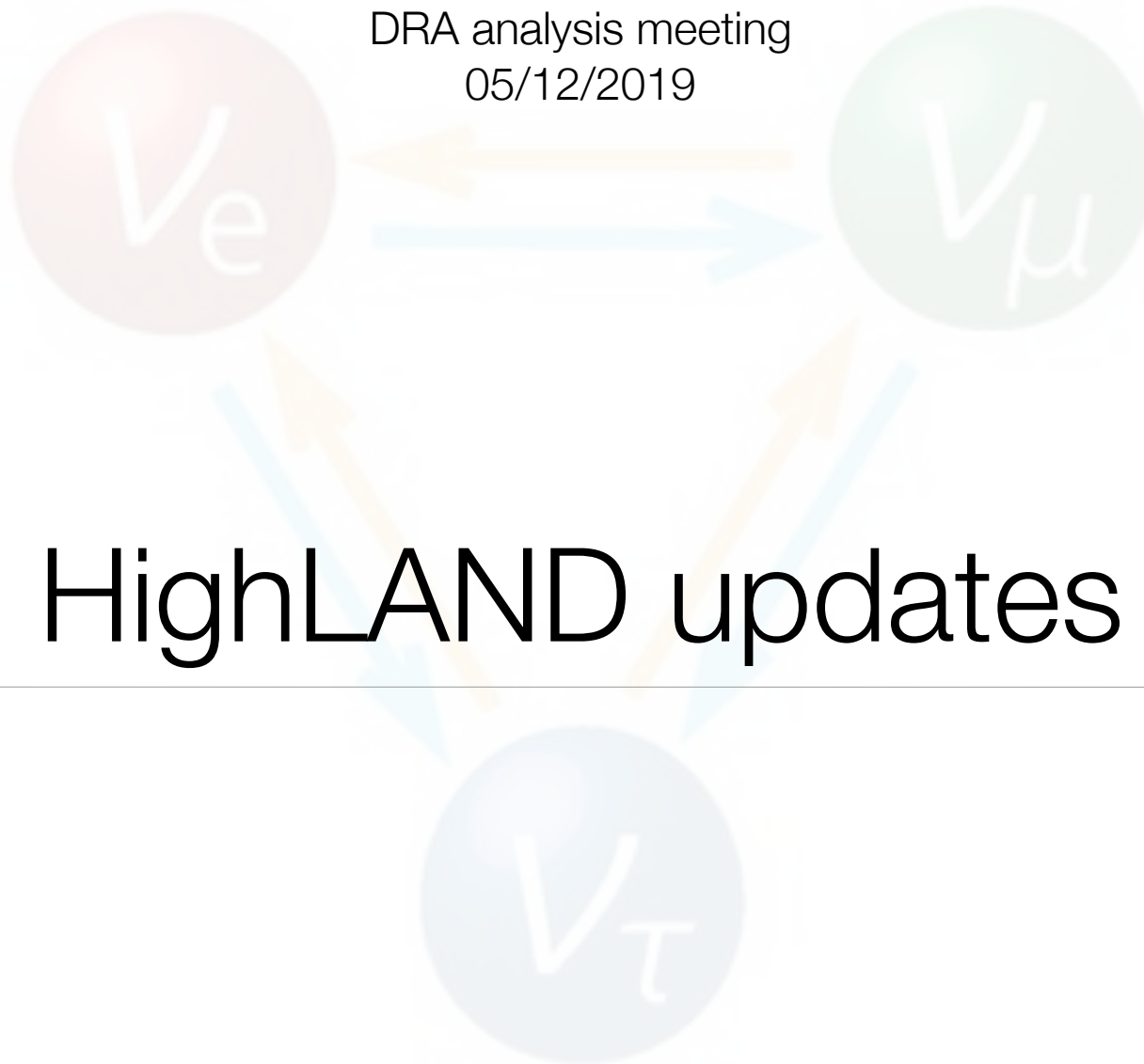# HighLAND updates

Anselmo Cervera Villanueva
IFIC-Valencia

HighLAND analysis framework

- **HighLAND: High Level Analysis Development**

- HighLAND has been crucial for T2K near detector analyses

- **Highly optimized, thread safe, compiled c++ code** and run on the shell command line (not as root macro)

- **Very compact set of packages**: 1 minute to download and 5 minutes to compile

- **Functionality**:

  - Event selection & systematics propagation

  - Drawing Tools & Event display

  - Data reduction

> **We have not started from scratch: All this functionality exists since long time from T2K. The system is fully validated !!!!**

DUNE — DEEP UNDERGROUND NEUTRINO EXPERIMENT

Previous HighLAND talks

- **MCC6** and **MCC7** eras

  - **FD sim/reco 23/11/2015**: https://indico.fnal.gov/conferenceDisplay.py?confId=10882
  - **LBL 24/11/2015**: https://indico.fnal.gov/conferenceDisplay.py?confId=10861
  - **S&C 15/12/2015**: https://indico.fnal.gov/conferenceDisplay.py?confId=11030
  - **DUNE CM, 14/09/2016**, https://indico.fnal.gov/event/10613/session/18/contribution/52/material/slides/0.pdf
  - **PD meas/ana 13/10/2016**: https://indico.fnal.gov/event/13110/
  - **DUNE CM 24/01/2017**: https://indico.fnal.gov/event/10641/session/12/contribution/81/material/slides/0.pdf

- **MCC11** era

  - ProtoDUNE analysis workshop 27/01/2019
    - https://indico.fnal.gov/event/19133/
    - Long talk explaining the framework and its functionality

  - DRA Analysis meeting 9/05/2019
    - https://indico.fnal.gov/event/20776/
    - Actual ProtoDUNE analyses using HighLAND

  - DUNE CM, 21/05/2019
    - https://indico.fnal.gov/event/18681/session/13/contribution/93/material/slides/0.pdf
    - Actual ProtoDUNE analyses using HighLAND

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT

Motivation of this talk

- Few students doing analysis in ProtoDUNE-SP are at a level such that they can start propagating systematics

- I was asked about the option of extracting the systematic propagation code from HighLAND and porting it to protoduneana

  - In my opinion this is a terrible mistake for several reasons

- I had several discussions with Georgios, which has expressed several concerns of the analysis group about HighLAND

  - In this talk I'll try to address those concerns

Requirements

- **Input files should be manageable:**

  - Small size

  - Fast to run over

  - Easy to understand for non-experts in simulation/reconstruction

- The analysis framework should be **decoupled** as much as possible from the much heavier **simulation/reconstruction** frameworks

  - Easy to install and compile

  - Easy to extend

  - Suitable for independent releases

- It should be possible to run the analysis and to do plots in a **laptop,** in Linux and MacOS

- It should be possible to run the analysis **without network connection**

DEEP UNDERGROUND
NEUTRINO EXPERIMENT

My concerns

- Doing analysis in LArSoft violates most previous requirements. But anyway, let's assume we go this way. This are my concerns:

    1. Porting not trivial at all. Many classes and concepts involved. It would require significant changes to the current code in HighLAND. **Validation in protoduneana will take a while**

    2. Systematics propagation should be very, very fast. **It will take a while to optimize the code in protoduneana**

    3. HighLAND produces a output tree with the results of the selection and systematics propagation, and has dedicated drawing tools. At plotting level one can: i) play with the cuts, ii) play with the systematics and change the binning. **Implementing this functionality in protoduneana will take time**

    4. At the end we want to address all previous points in protoduneana, so we will reinvent the wheel

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT

Analysis group's concerns

- In the next slides I address the concerns of the analysis group:

  - new framework to learn

  - need hit level info

  - highland uses CMT and not cmake

  - code sharing and git repository

Concern 1

- **Analyzers are familiar with protoduneana, not with HighLAND, and they will expend a lot of time trying to understand the new framework**

- It takes 5 minutes to download and compile HighLAND

- It is a very light framework, with very few concepts to understand. It will take a day for a student to start producing results

- Migrating an existing analysis to HighLAND will take few days. After that, progress will be much faster

- We can help with the migration !!!

Concern 2

- **All the analyses right now are still working at the hit level (vertex mis-reconstruction studies, elastic scattering tagging, machine learning PID and shower reconstruction, etc.)**

  - HighLAND can run on LArSoft reco files

- **This is highly complicated for protodune, an analyser has first to work with the reco files, then move to highland (or another analysis package) for selection and systematics and then move to a fit/unfolding package for a measurement**

  - Since HighLAND works with any input format (provided the appropriate converter) moving to higher analysis levels and reduced inputs is completely transparent

  - Ideally fitting/unfolding package should be such that it can accept any input (HighLAND or other)

Concern 3

- **HighLAND uses CMT while protoduneana uses cmake**

- Being HighLAND a separate set of packages it does not matter

- Anyway, we will migrate to cmake next week. It is straight forward. In fact it was already in the to do list

Concern 4

- **Analysers working on similar analyses, how they will commit and share code since the code lives on a different repository?**

- This is how we get existing highland packages

```
git clone https://next.ific.uv.es:8888/HighLAND/psychePolicy.git              psyche/psychePolicy/v0r0
git clone https://next.ific.uv.es:8888/HighLAND/psycheCore.git                psyche/psycheCore/v0r0
git clone https://next.ific.uv.es:8888/HighLAND/psycheUtils.git               psyche/psycheUtils/v0r0

git clone https://next.ific.uv.es:8888/duneHighLAND/psycheDUNEUtils.git       psyche/psycheDUNEUtils/v0r0
git clone https://next.ific.uv.es:8888/duneHighLAND/psycheEventModel.git      psyche/psycheEventModel/v0r0
git clone https://next.ific.uv.es:8888/duneHighLAND/psycheIO.git              psyche/psycheIO/v0r0
git clone https://next.ific.uv.es:8888/duneHighLAND/psycheSelections.git      psyche/psycheSelections/v0r0

git clone https://next.ific.uv.es:8888/HighLAND/highlandCore.git              highland2/highlandCore/v0r0
git clone https://next.ific.uv.es:8888/HighLAND/highlandTools.git             highland2/highlandTools/v0r0
git clone https://next.ific.uv.es:8888/HighLAND/highlandDoc.git               highland2/highlandDoc/v0r0

git clone https://next.ific.uv.es:8888/duneHighLAND/highlandEventModel.git    highland2/highlandEventModel/v0r0
git clone https://next.ific.uv.es:8888/duneHighLAND/highlandUtils.git         highland2/highlandUtils/v0r0
git clone https://next.ific.uv.es:8888/duneHighLAND/highlandCorrections.git   highland2/highlandCorrections/v0r0
git clone https://next.ific.uv.es:8888/duneHighLAND/LArSoftReader.git         highland2/LArSoftReader/v0r0
git clone https://next.ific.uv.es:8888/duneHighLAND/highlandIO.git            highland2/highlandIO/v0r0
git clone https://next.ific.uv.es:8888/duneHighLAND/baseAnalysis.git          highland2/baseAnalysis/v0r0
git clone https://next.ific.uv.es:8888/duneHighLAND/duneExampleAnalysis.git   highland2/duneExampleAnalysis/v0r0
git clone https://next.ific.uv.es:8888/duneHighLAND/protoDuneExampleAnalysis.git highland2/protoDuneExampleAnalysis/v0r0
```
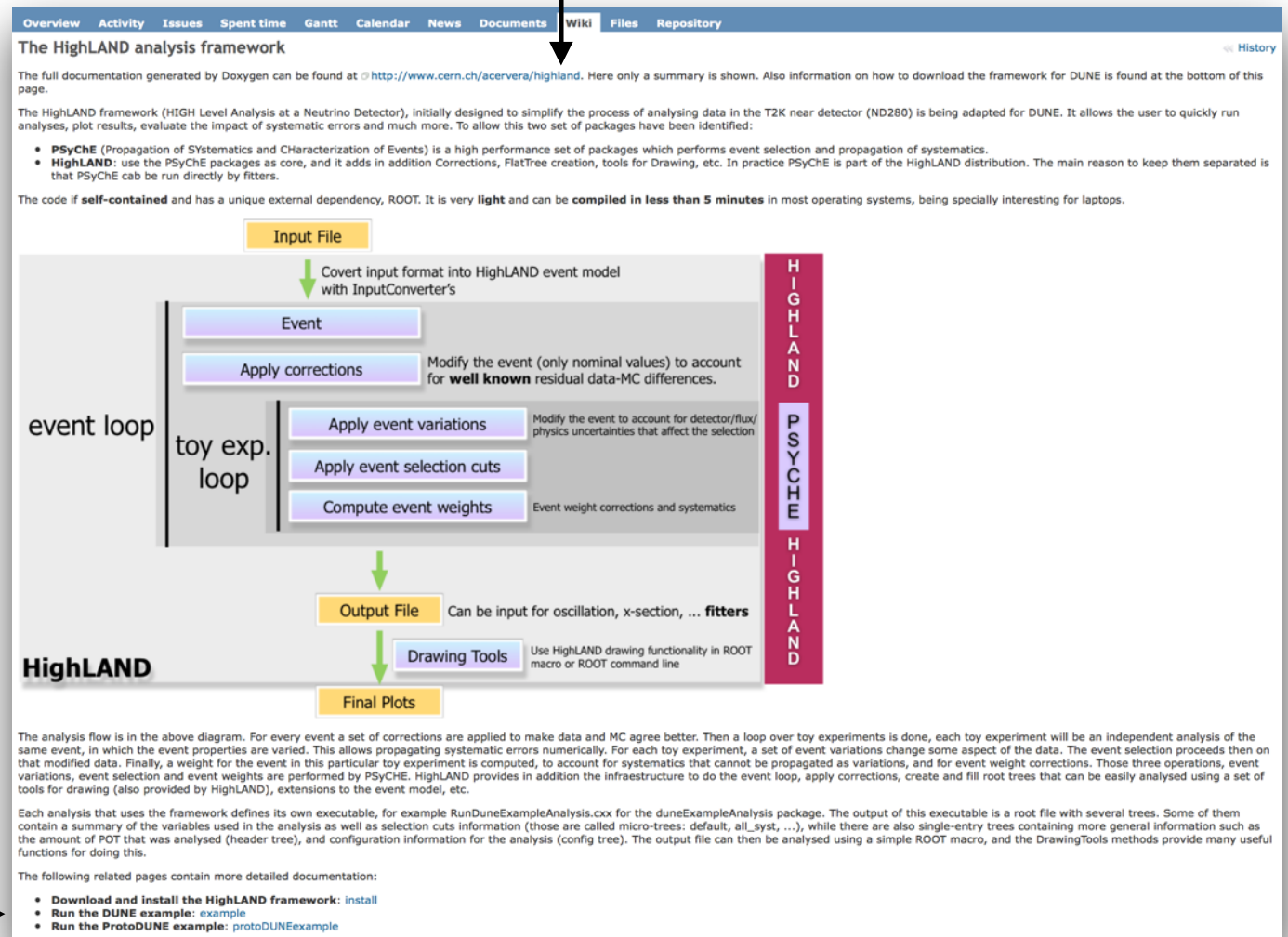
- We can get protoDUNE specific highland packages from the DUNE repository

```
git clone  http://cdcvs.fnal.gov/projects/protoduneHighland      highland2/myAnalysis/v0r0
```

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT

HighLAND in few hours

- This is the HighLAND redmine wiki page

  https://cdcvs.fnal.gov/redmine/projects/highland/wiki

link to detailed doxygen documentation

instructions and scripts
to install and run

Results in 10 minutes

- Download, compile and install the framework

https://cdcvs.fnal.gov/redmine/projects/highland/wiki/Install

The installation is done in few simple steps. First create a folder (i.e. HIGHLAND, or ANALYSIS) where you will put everything (CMT + HighLAND framework). Go inside that directory and save there the INSTALL.sh and setup.sh scripts that you can find at the bottom of this page. Or get them with wget

```
wget https://cdcvs.fnal.gov/redmine/attachments/download/51199/INSTALL.sh
wget https://cdcvs.fnal.gov/redmine/attachments/download/53882/setup.sh
```

Then just type:

```
source INSTALL.sh
```

5 minutes

- Run the ProtoDUNE example:

```
../Linux-x86_64/RunProtoDuneExampleAnalysis.exe -n 10000 -v -o output.root input.root
```

- Where input.root can be a LArSoft reco file or a HighLAND minitree, for example:

```
/dune/data/users/acervera/MiniTreesFilter/mini_data_run5387_calocorr_filter_pos_tof0-250.root
```
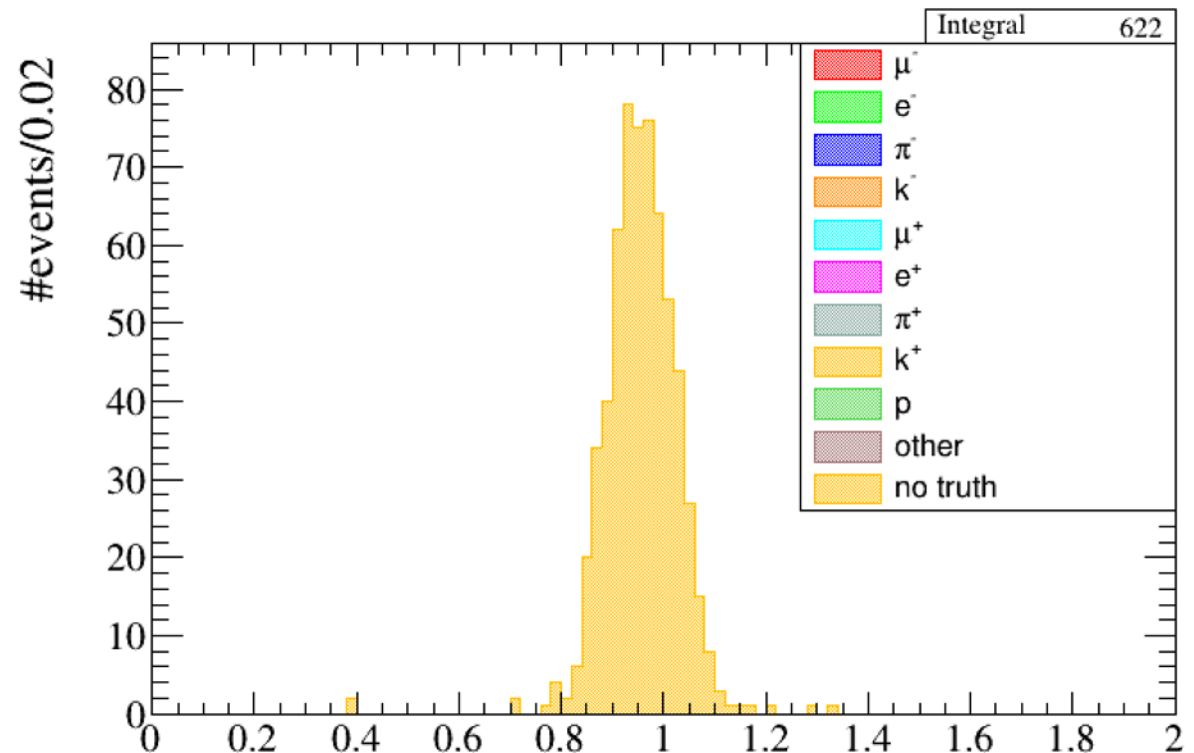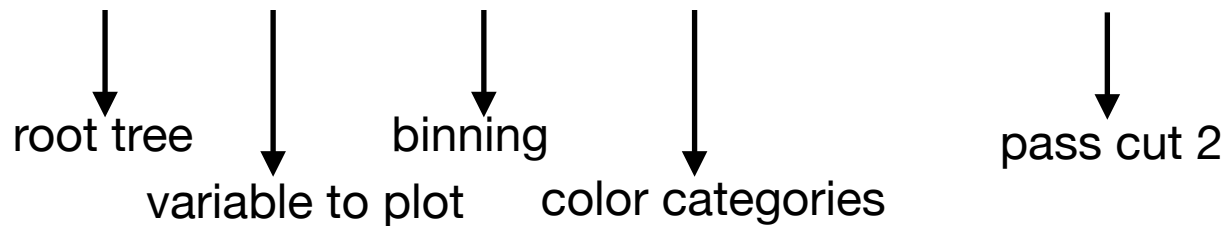
1 minute for 10K events

DU(VE DEEP UNDERGROUND NEUTRINO EXPERIMENT

Make a plot

```
root -l output.root

root [1] DrawingToolsBase draw("output.root")

root [2] TTree* d   = (TTree*)_file0->Get("default");


root [3] draw.Draw(d,"beam_mom_raw",100,0,2,"beamparticle","accum_level[0][0]>2")
```

root tree   binning   pass cut 2

variable to plot   color categories

HighLAND concepts

- **Event model:** Particle, Vertex, Hit, Beam, …, TrueParticle, etc

- **Event Selection** and associated **Steps** (cuts and actions)

- **The box**: an object to pass derived information from one step to another in the selection

- **Corrections:** for calibration and data/MC corrections mainly

- **Variations:** for systematic propagation

Stopping proton selection

beam track

TPC track

z=0

- Reproduce Heng-Ye's 1GeV/c stopping proton analysis
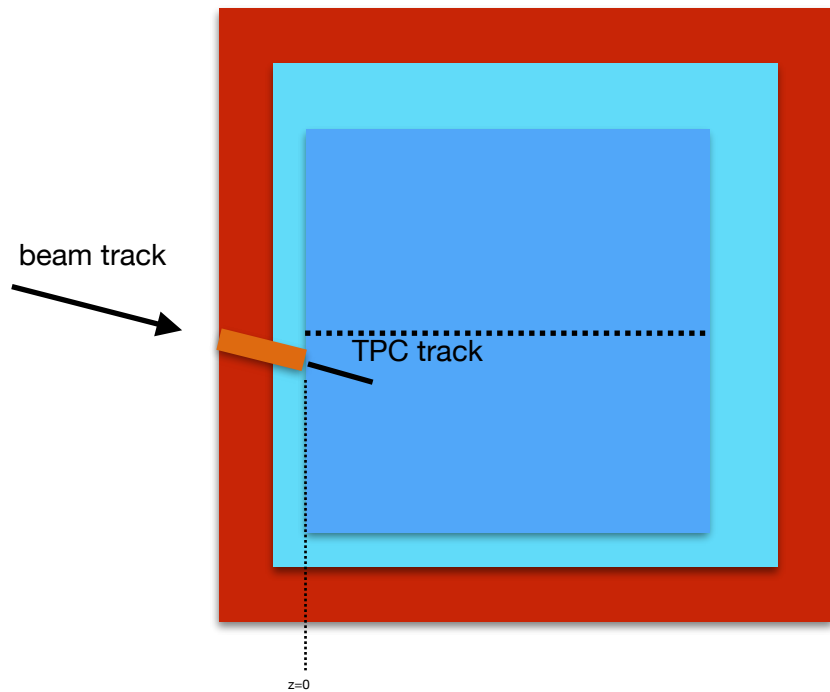
1. beam TOF compatible with proton
2. Δx, Δy at z=0
3. Δθ at z=0
4. Length/CSDA range (proton)

DU(N)E DEEP UNDERGROUND NEUTRINO EXPERIMENT
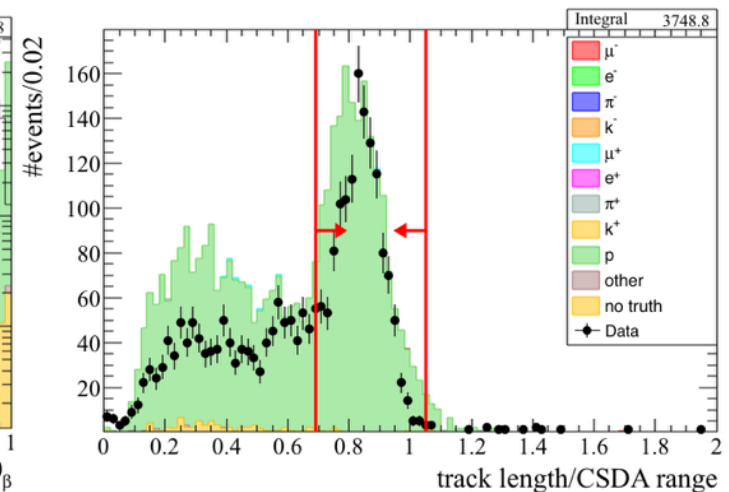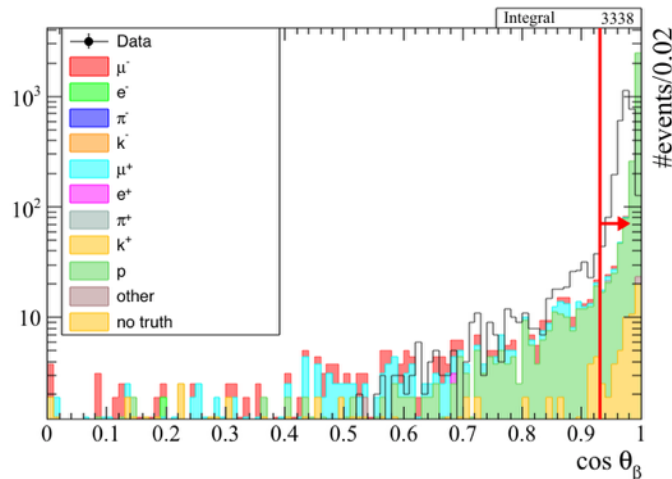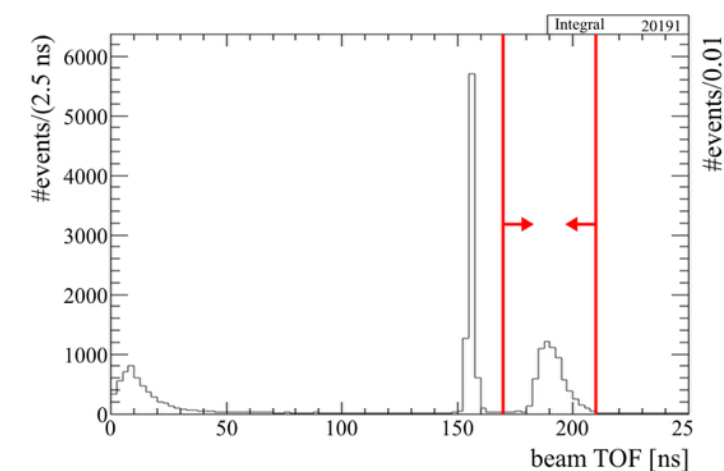
Event selection

- A selection is a collection of steps, which can be actions or cuts

```cpp
//*********************************************************************
void stoppingProtonSelection::DefineSteps(){
//*********************************************************************

  // Steps must be added in the right order
  // if "true" is added to the constructor of the step,
  // the step sequence is broken if cut is not passed (default is "false")
  AddStep(StepBase::kAction, "find main track",   new FindBeamTrackAction());
  AddStep(StepBase::kCut,    "beam protom",       new BeamProtonCut());
  AddStep(StepBase::kCut,    "beam track in TPC", new CandidateExistsCut());
  AddStep(StepBase::kCut,    "seltrk angle cut",  new BeamProtonAngleCut());
  AddStep(StepBase::kCut,    "proton CSDA range", new ProtonCSDARangeCut());

  SetBranchAlias(0,"trunk");
}
```

Example of action

```cpp
//********************************************************
bool FindBeamTrackAction::Apply(AnaEventC& event, ToyBoxB& boxB) const{
//********************************************************

  // The kaon candidate will be the most upstream track (lowest z stating position)

  // Cast the ToyBox to the appropriate type
  ToyBoxPD& box = *static_cast<ToyBoxPD*>(&boxB);

  // Get the array of tracks from the event
  AnaParticleB** tracks = static_cast<AnaEventB*>(&event)->Particles;
  int nTracks           = static_cast<AnaEventB*>(&event)->nParticles;

  // Get the beam particles
  AnaBeam* beam = static_cast<AnaBeam*>(static_cast<AnaEventB*>(&event)->Beam);
  if (!beam->BeamParticle) return true;

  // Loop over candidates
  Int_t ncand=0;
  for (Int_t i=0;i<nTracks; ++i){
    AnaParticle* part = static_cast<AnaParticle*>(tracks[i]);

    if (event.GetIsMC()){
      if (static_cast<AnaParticle*>(tracks[i])->Charge==-8888){
        box.MainTrack = static_cast<AnaParticle*>(tracks[i]);
        ncand++;
        break;
      }
    }
    // In data
    else{
      Float_t dx = part->PositionStart[0]-beam->BeamParticle->PositionEnd[0];
      Float_t dy = part->PositionStart[1]-beam->BeamParticle->PositionEnd[1];
      if (dx>-5 && dx<25 && dy>-10 && dy<10 && part->PositionStart[2]<100 && part->DirectionStart[2]>0.7 ){//&& part->Length>10){
        //    if (dx>-5 && dx<25 && dy>-10 && dy<10 && part->PositionStart[2]<100 && part->DirectionStart[2]>0.7 && part->Length>10){   // Cut for run 5210
        box.MainTrack = part;
        ncand++;
        break;
      }
    }
  }

  return true;
}
```

Example of cut

```cpp
//*******************************************************
bool BeamProtonCut::Apply(AnaEventC& event, ToyBoxB& boxB) const{
//*******************************************************

  (void)boxB;

  AnaBeam* beam = static_cast<AnaBeam*>(static_cast<AnaEventB*>(&event)->Beam);

  // In MC select true protons
  if (event.GetIsMC()){
    if (beam->BeamParticle){
      if (beam->BeamParticle->TrueObject)
        if (static_cast<AnaTrueParticle*>(beam->BeamParticle->TrueObject)->PDG==2212) return true;
    }
    return false;
  }
  // In DATA cut in beam TOF
  else{
    if (beam->TOF>_TOFMin && beam->TOF<_TOFMax) return true;
    else return false;
  }
}
```

Example of systematic

```cpp
//*************************************************************
void dEdxVariation::Apply(const ToyExperiment& toy, AnaEventC& event){
  //*************************************************************

  // Get the SystBox for this event
  SystBoxB* box = GetSystBox(event);

  // Loop over all relevant tracks for this variation
  for (Int_t itrk = 0; itrk < box->nRelevantRecObjects; itrk++){

    AnaParticle* part = static_cast<AnaParticle*>(box->RelevantRecObjects[itrk]);

    // The un-corrected particle
    const AnaParticle* original = static_cast<const AnaParticle*>(part->Original);

    if (!part->TrueObject)         continue; //?
    if (!original)                 continue; //?

    AnaTrueParticleB* truePart = static_cast<AnaTrueParticleB*>(part->TrueObject);

    // We need the errors part of the data file but as well the relative uncertainty for sigma
    Float_t mean_corr, mean_var;
    Int_t mean_index;

    // Note that the momentum changes if the mom resoltion, scale and bfield are also anabled.
    if (!GetBinValues(abs(truePart->PDG), mean_corr,  mean_var,  mean_index))  return;

    for (Int_t i=0;i<3;i++){
      for (Int_t j=0;j<part->NHitsPerPlane[i];j++){
        part->dEdx[i][j] = original->dEdx[i][j] *(1 +  mean_var*toy.GetToyVariations(_index)->Variations[mean_index]/mean_corr);
      }
    }
  }
}
```
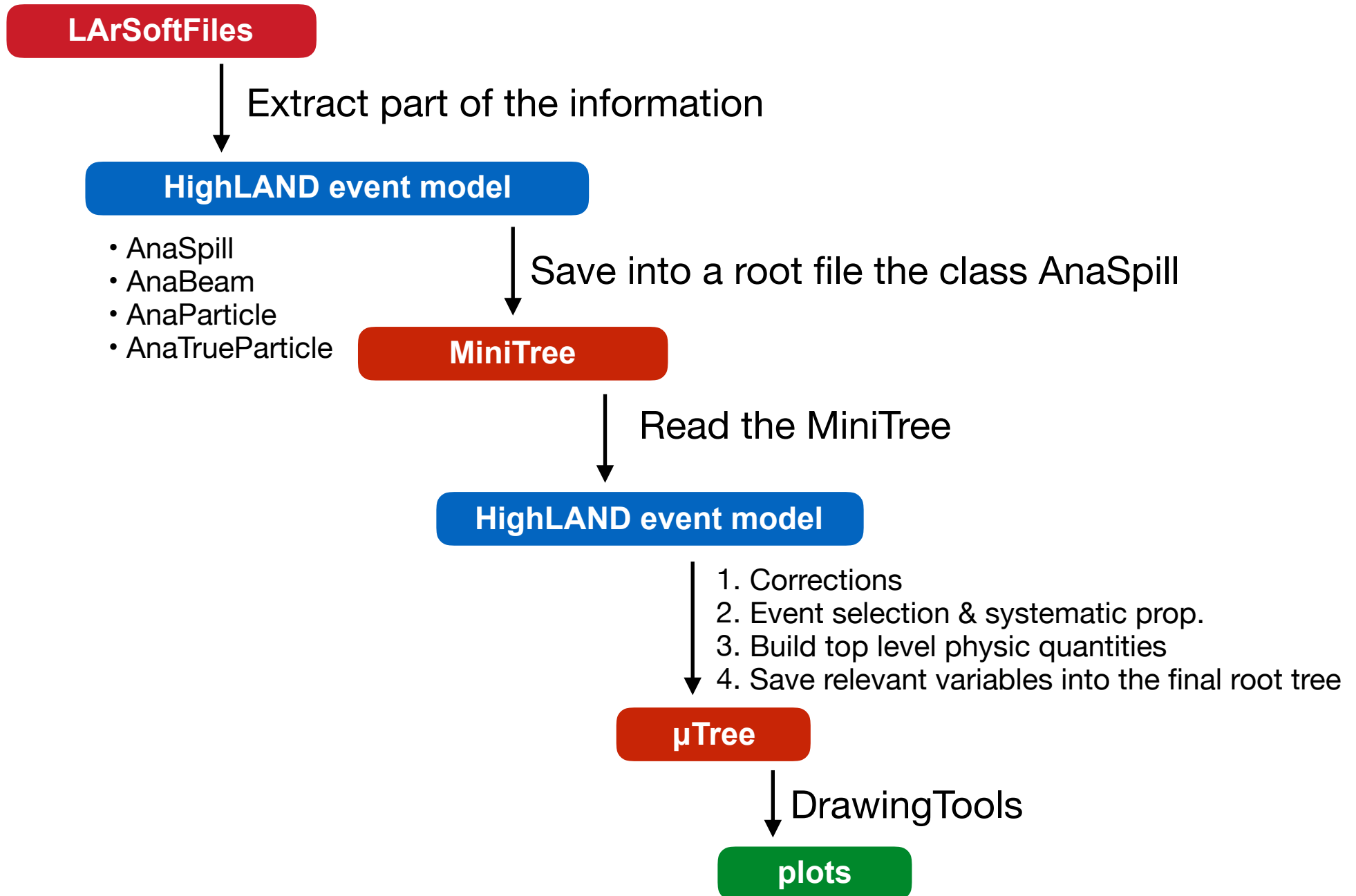
Analysis flow

Data reduction for MC

Example for 5225 LArSoft files: 1 GeV/c SCE

52250 events
13 TB

LArSoftFiles —— 10 days ——→ **μTree**

↓ 10 days

52159 events
160 GB

MiniTree-0 —— 3 h ——→ **μTree-0**

$-80 < X_0 < 20$
$370 < Y_0 < 470$

↓ 3 h

52114 events
36 GB

MiniTree-1 —— 1 h ——→ **μTree-1**

truth of saved recon particles and all its descendants

↓ 1 h

52114 events
8 GB

MiniTree-2 —— 3 min ——→ **μTree-2**

In a laptop

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT
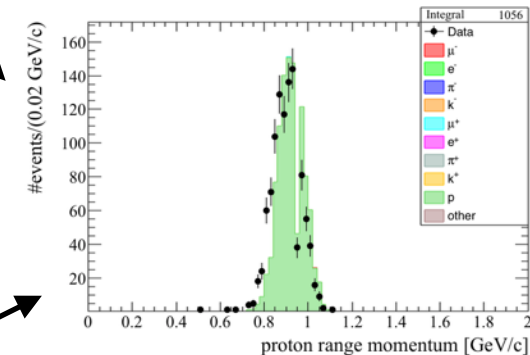
The path forward

- Get in contact with analyzers who want to start propagating systematics

  - Understand their needs

  - Make sure they want to explore the HighLAND option

  - Migrate their selections to HighLAND (I would be happy to do that).

- In parallel:

  - Migrate from CMT to cmake

  - Upgrade LArSoftConverter to ProtoDUNE production 2

- Give a tutorial at the analysis workshop at CERN