Run simulation→reconstruction chain and see the result

(based on materials from 31.10.2016, added example with CNN, SpacePoints and SWAN)

Start this excersises when you have:

- LArSoft environment configured

What is illustrated:

- run:
  - particle gun
  - G4 particle propagation through detector
  - detector readout simulation
  - reconstruction chain

  → using provided standard configuration files

- have a look into standard FHiCL's

- simulation and reconstruction results visualisation

Simulations / reconstruction in ProtoDUNE

particle gun

or

beam simulation

and/or

cosmogenics

Corsika

particle list, kinematics

general det.sim.

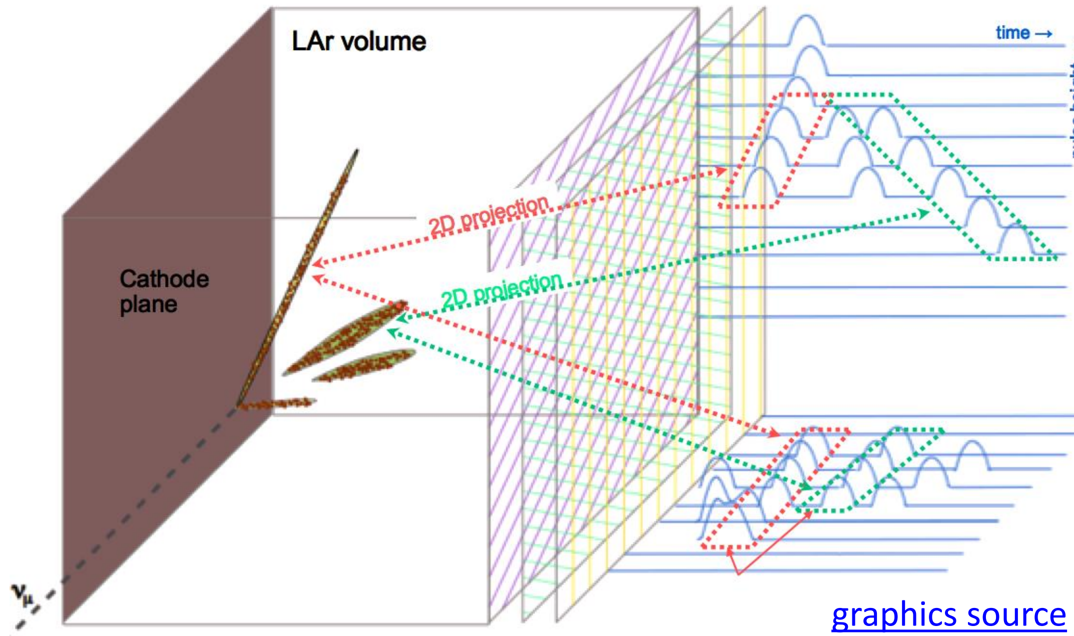
Geant4

energy depositions,
light

specific det.sim.

raw data like from real detector

reconstruction paths



graphics source

(re-)Login (just a remainder of setup steps)**, then prepare some directories for output:**

[rosulej@dune-vm-build-02 larsoft]$ **source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh**
Setting up larsoft UPS area... /cvmfs/fermilab.opensciencegrid.org/products/larsoft/
Setting up DUNE UPS area... /cvmfs/dune.opensciencegrid.org/products/dune/
[rosulej@dune-vm-build-02 larsoft]$ **source /cvmfs/fermilab.opensciencegrid.org/products/larsoft/setups**
[rosulej@dune-vm-build-02 larsoft]$ **setup mrb**
Use your release number here:
[rosulej@dune-vm-build-02 larsoft]$ **source localProducts_larsoft_v06_40_00_e14_prof/setup**

MRB_PROJECT=larsoft
MRB_PROJECT_VERSION=v06_40_00
MRB_QUALS=e14:prof
MRB_TOP=/afs/cern.ch/work/r/rosulej/larsoft
MRB_SOURCE=/afs/cern.ch/work/r/rosulej/larsoft/srcs
MRB_BUILDDIR=/afs/cern.ch/work/r/rosulej/larsoft/build_slf7.x86_64
MRB_INSTALL=/afs/cern.ch/work/r/rosulej/larsoft/localProducts_larsoft_v06_40_00_e14_prof
PRODUCTS=/afs/cern.ch/work/r/rosulej/larsoft/localProducts_larsoft_v06_40_00_e14_prof:/cvmfs/fermilab.opensciencegrid.org/products/larsoft:/cvmfs/dune.opensciencegrid.org/products/dune:/cvmfs/fermilab.opensciencegrid.org/products/larsoft:/cvmfs/fermilab.opensciencegrid.org/products/common/db

[rosulej@dune-vm-build-02 larsoft]$ **mrbslp**
local product directory is /afs/cern.ch/work/r/rosulej/larsoft/localProducts_larsoft_v06_40_00_e14_prof
----------- this block should be empty ------------------

[rosulej@dune-vm-build-02 ~]$ cd /afs/cern.ch/work/r/rosulej/ ← your folder, e.g. in AFS work area, or EOS in the future

[rosulej@dune-vm-build-02 rosulej]$ mkdir job ← make a directory for job configuration files

[rosulej@dune-vm-build-02 rosulej]$ mkdir data ← make a directory for LArSoft output files: sim, reco

[rosulej@dune-vm-build-02 rosulej]$ cd ~/larsoft ← **run LArSoft (lar -c …) from this location**
[rosulej@dune-vm-build-02 larsoft]$ ls
build_slf7.x86_64 ← **build area ($MRB_BUILD), run compilation from here**
localProducts_larsoft_v06_40_00_e14_prof
srcs ← **development area ($MRB_SOURCE), edit code here**

For your future work: **Upgrade to the newest LArSoft release** (please, login to a fresh terminal)**:**

[rosulej@dune-vm-build-02 larsoft]$ **source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh**
Setting up larsoft UPS area... /cvmfs/fermilab.opensciencegrid.org/products/larsoft/
Setting up DUNE UPS area... /cvmfs/dune.opensciencegrid.org/products/dune/
[rosulej@dune-vm-build-02 larsof]$ **source /cvmfs/fermilab.opensciencegrid.org/products/larsoft/setups**
[rosulej@dune-vm-build-02 larsof]$ **setup mrb**

Use your NEW release number here:
[rosulej@dune-vm-build-02 larsof]$ **mrb newDev -p -v v06_41_00 -q e14:prof**
[rosulej@dune-vm-build-02 larsof]$ **source localProducts_larsoft_v06_41_00_e9_prof/setup**

MRB_PROJECT=larsoft
MRB_PROJECT_VERSION=v06_41_00
MRB_QUALS=e14:prof
MRB_TOP=/afs/cern.ch/work/r/rosulej/larsoft
MRB_SOURCE=/afs/cern.ch/work/r/rosulej/larsoft/srcs
MRB_BUILDDIR=/afs/cern.ch/work/r/rosulej/larsoft/build_slf7.x86_64
MRB_INSTALL=/afs/cern.ch/work/r/rosulej/larsoft/localProducts_larsoft_v06_41_00_e14_prof
PRODUCTS=/afs/cern.ch/work/r/rosulej/larsoft/localProducts_larsoft_v06_41_00_e14_prof:/cvmfs/fermilab.opensciencegrid.org/products/larsoft:/cvmfs/dune.opensciencegrid.org/products/dune:/cvmfs/fermilab.opensciencegrid.org/products/larsoft:/cvmfs/fermilab.opensciencegrid.org/products/common/db

Update source codes in repositories you are developing (remember to call „kinit username@FNAL.GOV"):
[rosulej@dune-vm-build-02 larsof]$ **cd $MRB_SOURCE**
[rosulej@dune-vm-build-02 srcs]$ **cd dunetpc**
[rosulej@dune-vm-build-02 srcs]$ **git checkout develop**
[rosulej@dune-vm-build-02 srcs]$ **git pull**
…and rebuild:
[rosulej@dune-vm-build-02 srcs]$ **cd $MRB_BUILDDIR**
[rosulej@dune-vm-build-02 build_slf7.x86_64]$ **mrb z**
[rosulej@dune-vm-build-02 build_slf7.x86_64]$ **mrbsetenv**
local product directory is /afs/cern.ch/work/r/rosulej/larsoft/localProducts_larsoft_v06_41_00_e14_prof
----------- this block should be empty ------------------

[rosulej@dune-vm-build-02 build_slf7.x86_64]$ **mrb i –j4**

Run particle gun:

lar -c **srcs/dunetpc/fcl/protodune/gen/gen_protoDune_pion_2GeV_mono.fcl** -n 10 -o pi_gen.root

location of DUNE standard configuration files
you may skip entire path using .fcl files commited to the repository
(but it is good to know where they come from)
this will work as well:

lar -c gen_protoDune_pion_2GeV_mono.fcl **-n 10** -o pi_gen.root

number of events to produce (do not skip it, default is huge!)

lar -c gen_protoDune_pion_2GeV_mono.fcl -n 10 **-o pi_gen.root**

output file with events, place it in your data directory

What was generated:

- π^+ at 2GeV/c, monoenergetic
- in ProtoDUNE single-phase geometry
- in the position just in front of beam window, pointing towards it

Run G4 propagation of particles:

lar -c srcs/dunetpc/fcl/protodune/**g4/protoDUNE_g4.fcl** pi_gen.root -o pi_g4.root

↑

G4 (as each sim/reco step has its own directory for configuration files)

lar -c protoDUNE_g4.fcl **pi_gen.root** -o pi_g4.root

↑

use output file from previous step as input to the next step

lar -c protoDUNE_g4.fcl pi_gen.root **-o pi_g4.root**

↑

and put the path/filename for ouput

What was simulated:

- π^+ with the initial momentum 2GeV/c, all interactions of it and secondary particles
- in ProtoDUNE single-phase geometry
- full MC truth information is saved to the output file
- energy deposits in LAr were recorded as well

Run detector simulation:

lar -c srcs/dunetpc/fcl/protodune/**detsim/protoDUNE_detsim.fcl** pi_g4.root -o pi_detsim.root

here is the place of detector simulation job files

lar -c protoDUNE_detsim.fcl **pi_g4.root** -o pi_detsim.root

use G4 simulation as input to the detector simulation

lar -c protoDUNE_detsim.fcl pi_g4.root **-o pi_detsim.root**

and put the path/filename for ouput

What was simulated:

- scintillation light and ionization electrons from energy deposits in LAr
- recombination of ionization electrons
- attenuation of electrons number due to LAr impurities at given drift distance
- longitudinal and transverse diffusion of electrons at given drift distance
- ProtoDUNE single-phase readout:
    - E-field response for induction and collection planes
    - electronics response
    - noise
    - ADC waveforms on readout channels

Run reconstruction:

lar -c srcs/dunetpc/fcl/protodune/**reco/protoDUNE_reco.fcl** pi_detsim.root -o pi_reco.root

several reconstruction chains

lar -c protoDUNE_reco.fcl **pi_detsim.root** -o pi_reco.root

use detector simulation as input to the reconstruction

lar -c protoDUNE_reco.fcl pi_detsim.root **-o pi_reco.root**

and put the path/filename for ouput

What was reconstructed (with a success or not):

- deconvoluted ADC waveforms (recob::Wire's)
- hits on wire waveforms (recob::Hit)
- 2D clusters (recob::Cluster)
- 3D tracks (recob::Track)
- 3D vertices (recob::Vertex)
- Full event description (all above bounded in particle hierarchy, recob::PFParticle)

FHiCL job configuration file (here protoDUNE_reco.fcl):

```
#include "services_dune.fcl”
process_name: Reco
services: {…}                                   ← histogram root file configured
source: {…}                                     ← root input and max events
physics:
{
 producers:                                     ← module labels which will be used in this job
 {
  rns:         { module_type: RandomNumberSaver }
  caldata:      @local::producer_adcprep
  … … …
  pmtrack:      @local::dunefd_pmalgtrackmaker
 }
 reco: [ rns, ophit, opflash, … ]               ← sequence (path) of producer modules
 stream1:  [ out1 ]
 trigger_paths: [reco]                          ← actual selection of producer paths to tun
 end_paths:    [stream1]                        ← analyzers and output paths
}
outputs:
{ out1: {…} }
```

At the end of file: configuration of module parameters:

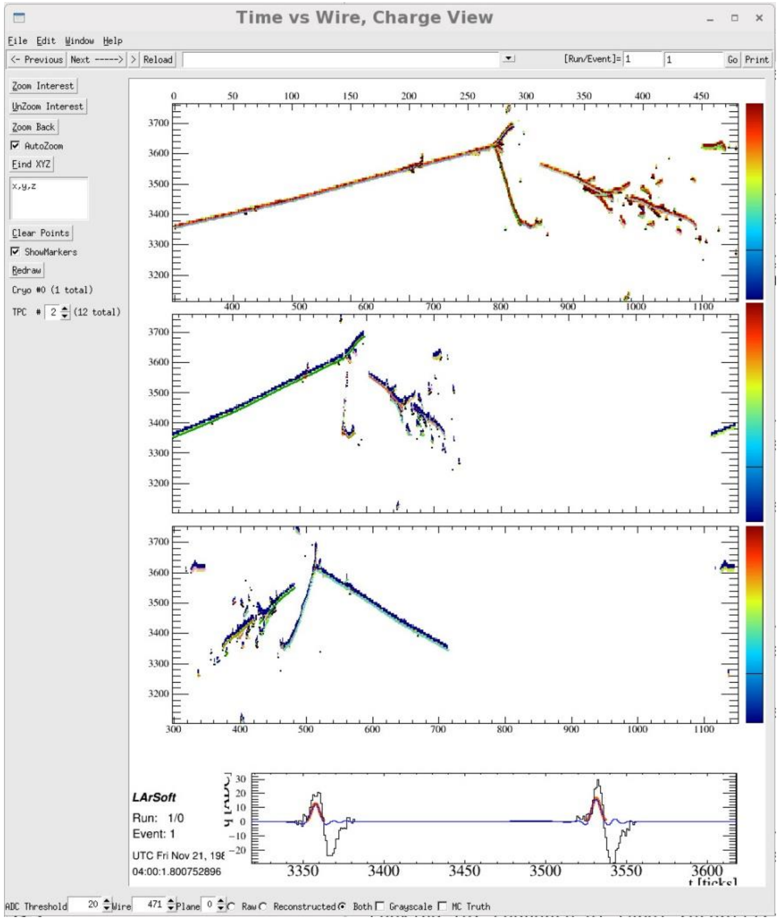
```
physics.producers.pmtrack.HitModuleLabel:          "linecluster"
physics.producers.pandora.HitFinderModuleLabel:   "linecluster”
…
```

Event display:

lar -c srcs/dunetpc/**dune/Utilities/evd_protoDUNE.fcl** pi_reco.root

or simply:

lar -c **evd_protoDUNE.fcl** pi_reco.root



All event display features illustrated in **YoungDUNE tutorial**:
https://indico.fnal.gov/getFile.py/access?contribId=5&resId=0&materialId=slides&confId=12889

SpacePoints, CNN feature labeling, SWAN notebook

Run reconstruction:

lar -c sp_and_cnn_reco_job.fcl pi_detsim.root -o pi_reco.root -T testspcnn.root

↑

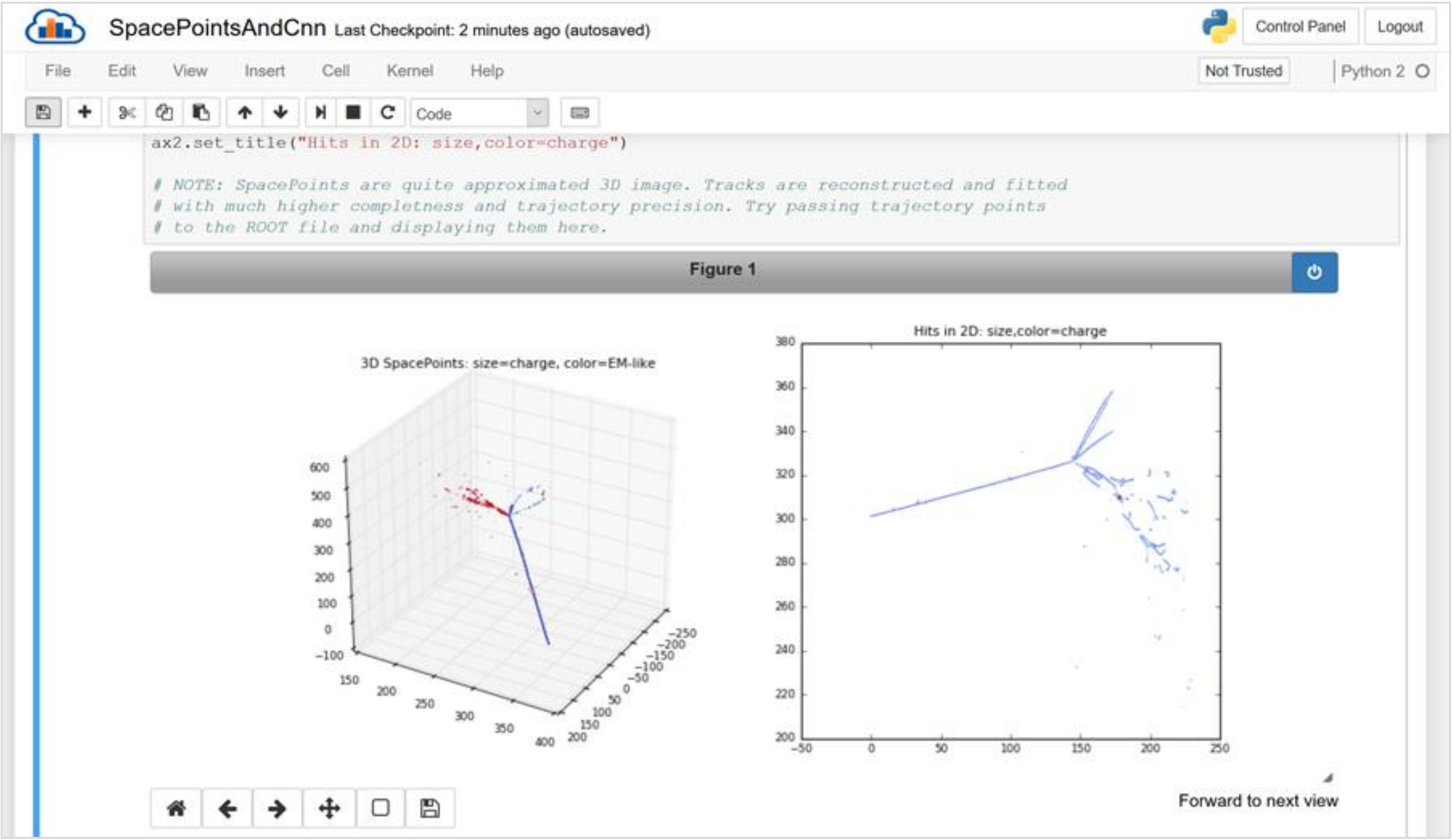
another reco chain plus analyzer module dumping information to a plain ROOT file

Reconstructed here:

- deconvoluted ADC waveforms (recob::Wire's)
- hits on wire waveforms (recob::Hit)
- 2D clusters (recob::Cluster)
- 3D tracks (recob::Track)
- 3D vertices (recob::Vertex)
- 3D space points
- CNN EM/track labeling

Analyzer module:
https://cdcvs.fnal.gov/redmine/projects/dunetpc/repository/revisions/develop/entry/dune/Protodune/TutorialExamples/ReadSpacePointAndCnn_module.cc

Results accessed from SWAN



https://cernbox.cern.ch/index.php/s/oFHCuoXLcZemGxA