

## Setup and Running of Pandora ND

29/3/2022

### Create a setup file

Make sure your computer can use CVMFS.

Create a file called "setup.sh". It should look something like:

“””

```
source /cvmfs/dune.opensciencegrid.org/products/dune/setup_dune.sh
```

```
export LIBGL_ALWAYS_SOFTWARE=1
```

```
export GALLIUM_DRIVER=softpipe
```

```
export DRAW_USE_LLVM=0
```

```
setup gcc v8_2_0
```

```
setup git v2_20_1
```

```
setup python v2_7_13d -f Linux64bit+3.10-2.17
```

```
setup eigen v3_3_9a
```

```
setup root v6_18_04d -q e19:prof
```

```
setup geant4 v4_10_6_p01c -q e19:prof
```

```
setup cmake v3_19_6
```

```
setup genie v3_00_06h -q e19:prof
```

```
setup genie_xsec v3_00_04 -q G1810a0211a:e1000:k250
```

```
export MY_TEST_AREA=`pwd`
```

“””

Run *source setup.sh*

### Getting edep-sim

```
git clone https://github.com/ClarkMcGrew/edep-sim
```

```
cd edep-sim
```

```
source setup.sh
```

```
export CMAKE_PREFIX_PATH=${EDEP_ROOT}/${EDEP_TARGET}
```

```
mkdir -p install
```

```
cd build
```

```
cmake -DCMAKE_INSTALL_PREFIX=/YOUR/PATH/edep-sim/install ..
```

```
make -j4 install
```

Note – if it is complaining, try running this separately (*setup geant4 v4\_10\_6\_p01c -q e19:prof*), then make again. Just push through if it lets you, it seems to have an issue with finding geant.

```
export LD_LIBRARY_PATH=/YOUR/PATH/edep-sim/install/lib:$LD_LIBRARY_PATH
```

```
export PATH=/YOUR/PATH/edep-sim/install/bin:$PATH
```

I would then add these to your setup.sh

### Getting Pandora

Follow #2 at this link: <https://github.com/PandoraPFA/Documentation> in the area where your setup script is

If a command says it can't find something, try running a second time, and normally it will find it

Then add this to your setup script:

```
“””
```

```
export MY_TEST_AREA=`pwd`  
cd YOUR/PATH/LArMachineLearningData/PandoraMVADData  
export FW_SEARCH_PATH=$FW_SEARCH_PATH:`pwd`  
cd ../PandoraMVAs  
export FW_SEARCH_PATH=$FW_SEARCH_PATH:`pwd`  
cd ../PandoraNetworkData  
export FW_SEARCH_PATH=$FW_SEARCH_PATH:`pwd`  
cd ../../LArReco/settings  
export FW_SEARCH_PATH=$FW_SEARCH_PATH:`pwd`  
cd $MY_TEST_AREA  
“””
```

Now get the right branch:

```
cd LArReco  
git checkout remotes/origin/feature/edep-reco  
mkdir build  
cd build  
cmake -DCMAKE_MODULE_PATH="$MY_TEST_AREA/PandoraPFA/cmakemodules;  
$ROOTSYS/etc/cmake" -DPANDORA_MONITORING=ON -  
DPandoraSDK_DIR=$MY_TEST_AREA/PandoraSDK/ -  
DPandoraMonitoring_DIR=$MY_TEST_AREA/PandoraMonitoring/ -  
DLArContent_DIR=$MY_TEST_AREA/LArContent/ ..  
make -j4 install
```

This should now be setup to run. May need to delete and remake the build folder after changing branch.

### **Get some files to work with**

You should know which ones you want to work with, but we shall use some from here as an example: /pnfs/dune/persistent/users/jback/EdepSimFiles/particles\_UniBox/

### **Run**

An example of a run command, given in the LArReco dir is:

```
./bin/PandoraInterface -i settings/PandoraSettings_EDepReco.xml -r AllHitsCR -j LArTPC -N -n  
1000 -e muPlus_0p1GeV.root
```

You can handscan with the file as is, or remove visual monitoring to run through all events. You can use the -r option to change between the cosmic and neutrino hypotheses with *allhitscr* and *allhitsnu*

### **Validation**

This running should have produced a file called Validation.root

If any histogram parameters need to be adjusted, this can be done in validation/Validation.C

Now produce a file containing needed histograms; choose the ones you want

```
root
```

```
TFile *f2 = new TFile("ValidationHistograms.root", "CREATE")
```

```
.L /usera/afm67/2021/September/NDLAr_Day1_Testing/LArReco/validation/Validation.C+
```

Parameters p

```
p.m_testBeamMode=true (true for test beam, false for cosmic)
p.m_applyUbooneFiducialCut=false
p.m_histogramOutput=true
p.m_displayMatchedEvents=true
Validation("Validation.root", p)
ALL_INTERACTIONS_MUON_HitsEfficiency-
>Write("ALL_INTERACTIONS_MUON_HitsEfficiency")
ALL_INTERACTIONS_MUON_MomentumEfficiency-
>Write("ALL_INTERACTIONS_MUON_MomentumEfficiency")
ALL_INTERACTIONS_MUON_Completeness-
>Write("ALL_INTERACTIONS_MUON_Completeness")
ALL_INTERACTIONS_MUON_Purity->Write("ALL_INTERACTIONS_MUON_Purity")
ALL_INTERACTIONS_VtxDeltaX->Write("ALL_INTERACTIONS_VtxDeltaX")
ALL_INTERACTIONS_VtxDeltaY->Write("ALL_INTERACTIONS_VtxDeltaY")
ALL_INTERACTIONS_VtxDeltaZ->Write("ALL_INTERACTIONS_VtxDeltaZ")
ALL_INTERACTIONS_VtxDeltaR->Write("ALL_INTERACTIONS_VtxDeltaR")
This will now have produced and filled ValidationHistograms.root
```

Note – if you wish to view the files within ROOT, you can use  
<HIST\_NAME>->Draw("hist")

### **Viewing the Histograms**

You can then either write your own file to view and arrange the histograms as you please from the ValidationHistograms.root file, or if you wish to overlay them like I have done, you can use overlayHistograms.cc. That can currently be found here:  
<https://github.com/afm1g15/LArReco/blob/feature/edep-reco-alex/overlayHistograms.cc>

To run:

```
root -l
.x overlayHistograms.cc("VtxDeltaR", "ValidationHistograms_0p1.root",
"ValidationHistograms_0p2.root", "ValidationHistograms_0p3.root",
"ValidationHistograms_0p4.root", "ValidationHistograms_0p5.root",
"ValidationHistograms_0p6.root", "ValidationHistograms_0p7.root",
"ValidationHistograms_0p8.root", "ValidationHistograms_0p9.root",
"ValidationHistograms_1p0.root", "ValidationHistograms_1p25.root",
"ValidationHistograms_1p5.root", "ValidationHistograms_1p75.root",
"ValidationHistograms_2p0.root", "ValidationHistograms_2p5.root",
"ValidationHistograms_3p0.root", "ValidationHistograms_4p0", "0p1 GeV", "0p2", "0p3", "0p4",
"0p5 GeV", "0p6", "0p7", "0p8", "0p9", "1p0", "1p25", "1p5 GeV", "1p75", "2p0", "2p5", "3p0",
"4p0 GeV")
```

The large number of commands is there if one wants to overlay every single generated histogram. However, for many of the histograms (e.g. VtxDeltaR) I have just been overlaying four energies (0.1GeV, 0.5GeV, 1.5GeV, 4.0GeV) and filling the rest with dummy variables.

This file will do basic cosmic changes, but anything more requires going into Validation.C and re running.