Adding Pandora to the LArSoft CI

Ryan Cross

Summary

1. Quick Overview of the Existing Infrastructure

2. New specific FD-Pandora bits

3. Ongoing Work

Existing CI - What

There has been a continuous integration (CI) system for a decently long time, so you've maybe seen bits around it, but I thought I'd give a small overview of what a CI system is, then specifically how it applies here.

All of this has been in place for a while, and I'll be discussing how I added a specific config for Pandora, but its only a part of this existing system.

Broadly, a CI system is some form of automated workflow that runs a configured set of jobs over your code base, checking and reporting the results.

These jobs can be really simple ones, or fairly complex:

- Check the code compiles
- Run test suites
- Regression Tests

- Syntax Checks
- Run Performance Checks
- Run full test workflows

Here, we are most interested in the last one, where we can have a preconfigured set of `fcl` jobs run on the grid, then produce some plots to check everything looks as expected.

Existing CI - How

For the LArSoft CI, there is the lar_ci repo, which lays out the config files for the various workflows, across a few experiments, as well as a comprehensive wiki on running it on demand.

These config files just list out what FCL files should be run, where stuff should be stored, other scripts to run, and any job parameters.

With this, its easy enough to have the CI run a few FCL jobs to get some analysis output, `hadd` the results, and then run a ROOT macro to produce some plots, which will appear on the CI dashboard.

```
...

[reco]
FHiCL               = standard_reco_dune10kt_nu_1x2x6.fcl,select_ana_dune10kt_nu.fcl,standard_ana_dune10kt_1x2x6.fcl
grid_usage_model    = DEDICATED,OPPORTUNISTIC,OFFSITE
expected_lifetime   = 300m
memory              = 2000MB
disk                = 9GB
executable          = lar
arguments           = --rethrow-all
input_from_stage    = sim
input_filename      = prodgenie_nue_dune10kt_1x2x6_detsim.root
output_filename     = prodgenie_nue_dune10kt_1x2x6_ana.root
output_to_transfer  = ana_hist.root,prodgenie_nue_dune10kt_1x2x6_ana.root

...
```

Existing CI - Where

There is then a Web dashboard at http://lar-ci-history.fnal.gov/LarCI/app which can be used to see how ongoing or historic CI jobs. This is for a bunch of experiments, but includes DUNE.

You'll see something like this:

This shows a few of the steps (setting up, getting the code, building, small tests, CI tests). Hovering over the build number on the left will show why it was kicked off, and I'll go through that in a moment.

Clicking on any section we can see it in more detail. Some jobs just list how often they pass fail, others will give warnings etc for performance regressions. The more interesting part for this talk is the "CI Validation" jobs, that actually run a large number of events, (whereas "CI Tests" only kick a few jobs off to test it doesn't die).

For these larger jobs, you can get full plot outputs, including comparison plots and plots of chi-squared values over time and more. I'll go through that when showing the Pandora bits.

Existing CI - When

The final main point to mention about the existing CI infrastructure, is when/how it is run.

The most common cause for it to be run, is due to commits/pushes to the `develop` branch of `dunetpc`, but it can be kicked off for other reasons, like LArSoft changes kicking off a run to test changes upstream aren't causing issues for the various experiments.

To my knowledge, the more comprehensive, plot making, jobs are kicked off manually. That can be by anyone (following the steps in the lar_ci wiki!), or periodically its run for new releases etc.

At least for the Pandora workflow, we expect to run it when we are reviewing work, since the command you give can resolve against remote repos, to test things before it is merged.

```
T: trigger script by rcross@dunegpvm15.fnal.gov - revisions: PandoraPFA/LArContent>larpandoracontent@feature/larpandoracontent_v03_22_08
W: CI_VALIDATION_DUNE
P: mrb
```

Pandora CI

The first thing to address is what/why a Pandora specific workflow for the CI.

Broadly, this was driven by a desire to decouple results away from other modules in the reconstruction chain, as well as adding more reconstruction-focused plots.

The existing CI runs the full chain, sim through to reco, which means changes are harder to disentangle. Is a change in the results because Pandora changed? or perhaps some module upstream did? This means it is harder to make an informed decision.

Secondly, a lot of the existing plots are more physics focused, whereas its useful for Pandora to have more reconstruction focused plots. For development we want things like:

- Completeness/Purity,
- Cluster numbers
- Angular distributions
- Lengths/Positions of particles

Pandora CI

These two desires have two main differences to the existing CI workflows. I made these changes with a lot of help from Vito, to fit Pandora's needs:

- The Pandora configuration uses partially processed files. We run the standard simulation steps, then the first part of the reco, and then save those files.
- We use a dedicated reconstruction analysis module, written by Maria Brunetti (Warwick), to plot out more reconstruction specific plots.

The pre-processed files use the normal sim/reco path, but it stops just before Pandora. These files are then saved, and used in each CI run.
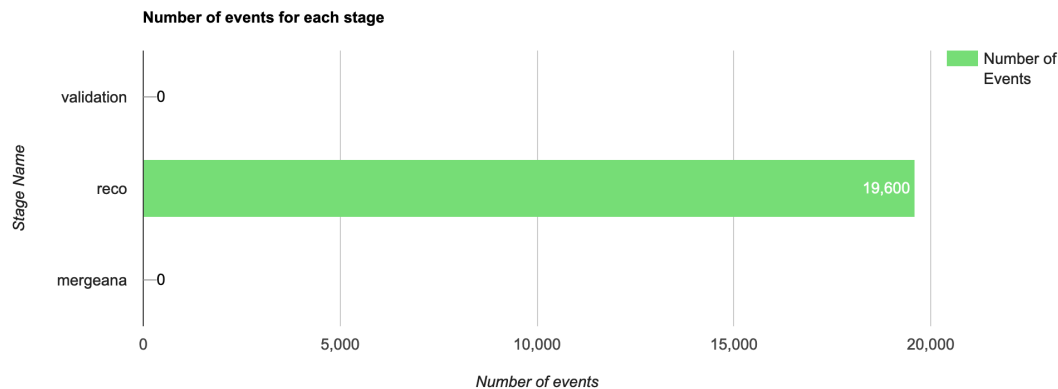
Pandora CI

A recent run of this new workflow can be found here. You should be able to browse the results produced using that link.

As a short disclaimer, the plots shown in the CI shouldn't be taken as fully demonstrative of Pandora's performance for Physics analysis. The plots are tailored to help profile the reconstruction, rather than show the absolute performance of Pandora.

I will highlight a few of the key features though.

```
+pandoraAnaEff/
+pandoraAnaEffCompare/
```

TEXT
CI
Validation
Report
ComparisonChiSquare.txt

ComparisonChiSquare

TEXT
CI
Validation
Report
overall_efficiency.txt

overall_efficiency

At the top, there is two folders of plots. The first is the raw output of the CI job, and the second contains comparison plots versus some tagged LArSoft version.

At the bottom, there is two files, that contain the results of the plots, but in numeric form. That is efficiencies for the raw results, and a comparison for the reference plots.

An example

nPFParticles
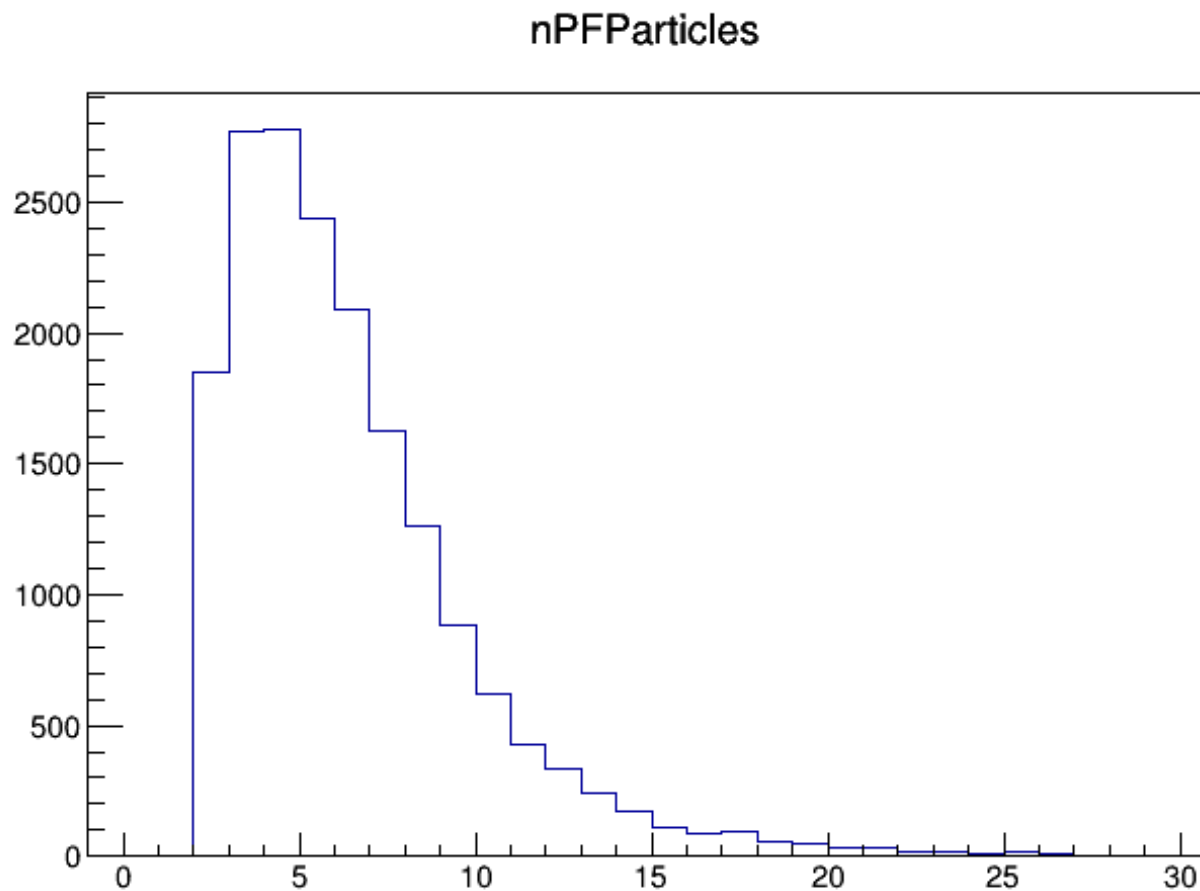
An example
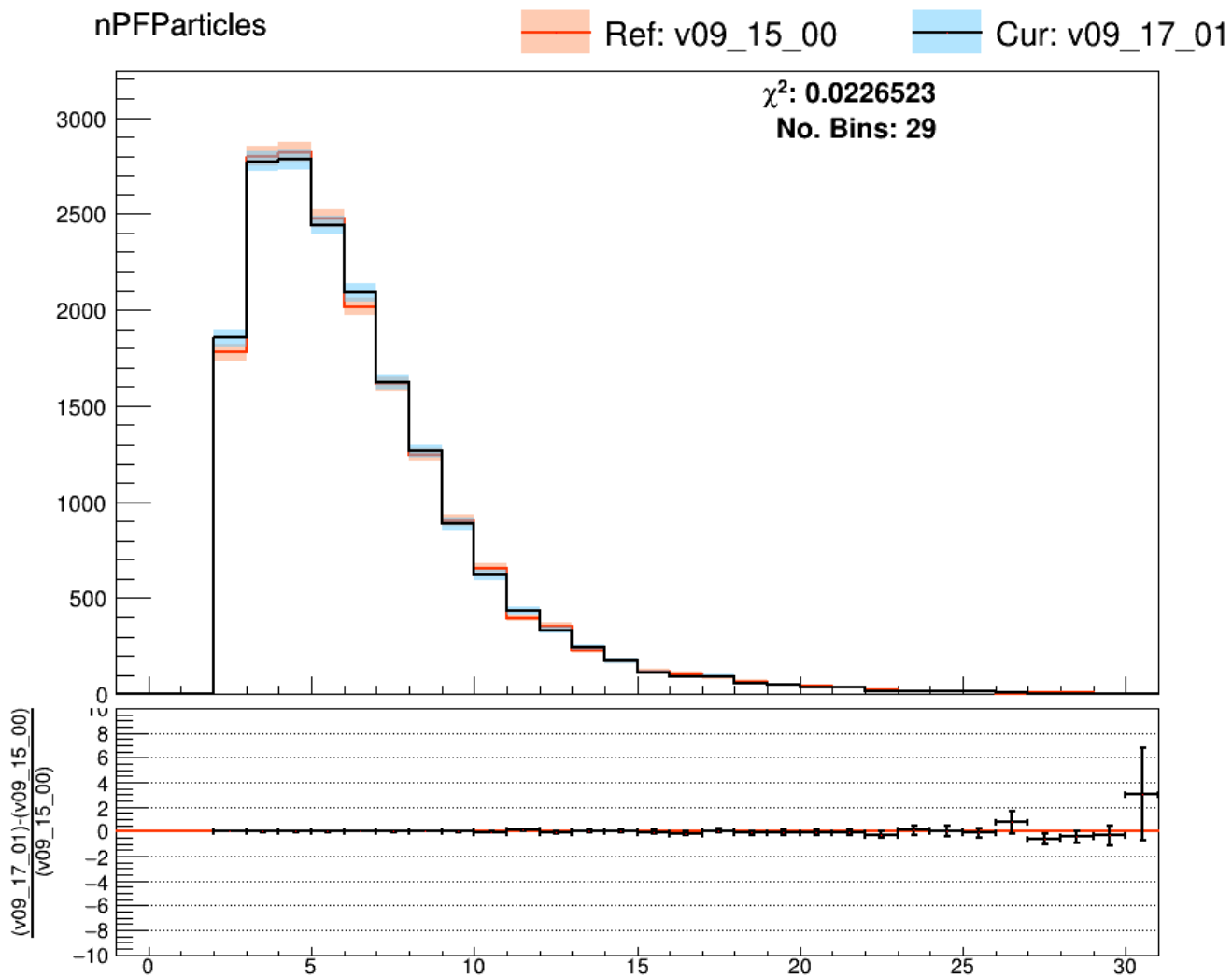
Ongoing Work

There is still one or two bits I'd like to add to this workflow:

- A better "apples to apples comparison". That is, the current workflow always compares against a reference file of 20,000 events, even if a few jobs failed[*]. It would be nice to always compare the same events, so select a subset of those 20,000 for the comparison.

- Expand the plots to include some form of selection. Whilst the main objective is to see reconstruction metrics and how they change, it can be useful to see how Physics results change. Adding in Dom's selection would allow us to see how that is affected by changes to the reconstruction.

    - i.e. If we improve shower reco, how much does that improve the efficiency at the end, or does some change alter the background we expect etc.

Jobs can fail for various reasons, unrelated to the actual changes being tested.

Summary

In summary:

- There is already a comprehensive CI suite, that is run periodically, and can be used to keep an eye on the results of the simulation/reconstruction chain.
- If there is things missing, we should try and add it! Since it uses all the existing infrastructure (Analyser modules, ROOT macros etc.) it shouldn't be too much work to extend an existing workflow with a new plot, or add a new config entirely.
- Pandora now has a CI workflow for the DUNE FD (almost merged) which should allow reconstruction changes and the effect they have to be more easily tracked.
- Further changes will make this comparison more robust, and extend it beyond reconstruction metrics only.

Adding Pandora to the LArSoft CI

Ryan Cross

Backup Slides