



External building of consolidated Pandora events

Andrew Smith for the Pandora team

LArSoft coordination meeting - 14th August 2018

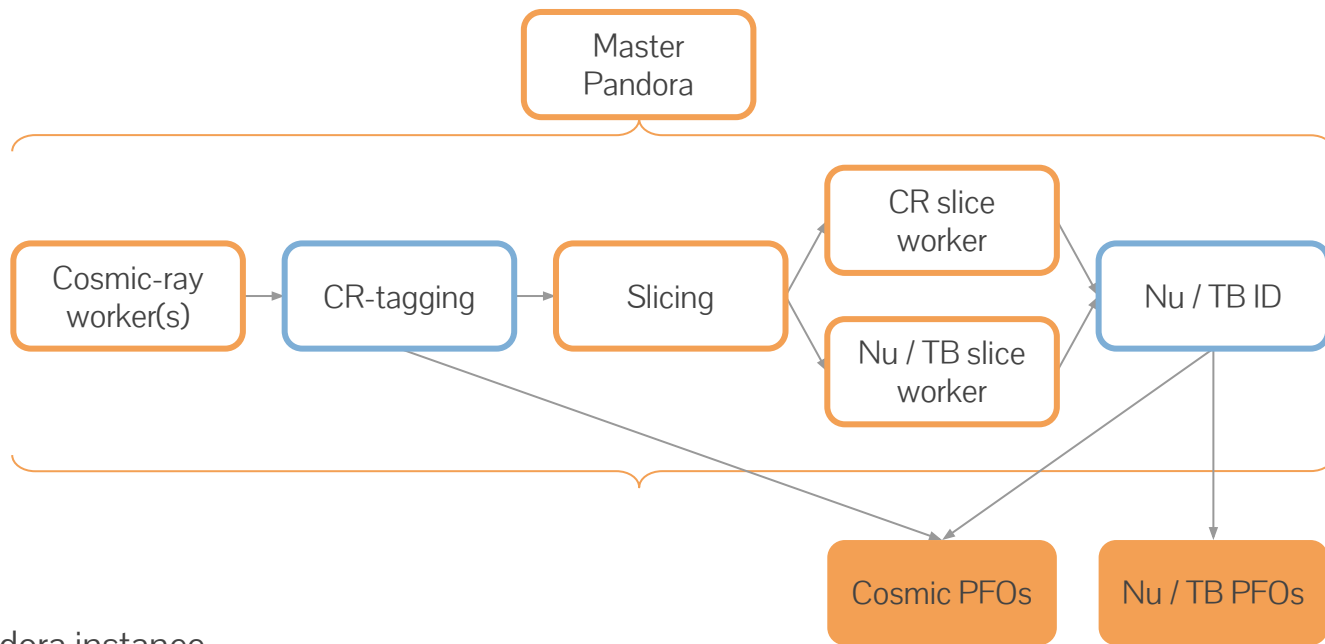
Overview of recent updates




- Refactoring of LArPandoraOutput
 - Responsible for the conversion of Pandora outputs to LArSoft products
- Extension of LArPandoraOutput to optionally persist alternate patrec outcomes
 - Allows downstream users to modify steering decisions made in Pandora
- New “external event building” module (optional)
 - Provides the framework to easily use the alternate patrec outcomes with safety against double counting, etc.
- New “SimpleNeutrinoIdTool” (optional)
 - A simple example of the logic required to build external events. Reproduces the consolidated output built in Pandora

Motivation for external event building

- Pandora needs to make several important steering decisions to deliver a consolidated output
 - Importantly, neutrino / test-beam particle ID
 - Currently, decision is made inside Pandora using one of two algorithms optimised for the **MicroBooNE** and **protoDUNE** use-cases
- Although decisions are made extremely carefully, only TPC information is available to Pandora
 - We want to persist access to these decisions outside of Pandora, so users can optionally choose to develop their own neutrino / test-beam particle ID using other information (PMT / CRT)

Pandora's consolidated algorithm flow



-  Pandora instance
-  Pandora algorithm
-  Pandora collection

Nu = Neutrino
TB = Test-beam

Persisting more information to LArSoft

- When switched on, the new code will output:

- pandoraPatRec
 - Standard consolidated output: 1 nu / M TB + N cosmics

- pandoraPatRec:allOutcomes
 - Persist all possible reconstruction outcomes:
 - Clear cosmics
 - Slices reconstructed under both neutrino / TB and cosmic hypothesis

Standard use-case

Communicates only with master Pandora instance

Users can ask for the ID'd PFFParticle using PDG code or metadata

Take the decisions made by Pandora at face-value

Persisting more information to LArSoft

- When switched on, the new code will output:
 - pandoraPatRec
 - Standard consolidated output: 1 nu / M TB + N cosmics

“Advanced” use-case

Communicates with master + daughter worker Pandora instances

Use the new ExternalEventBuilding module in LArSoft to look at both outcomes of each slice and decide which (if any) contains the nu / TB

- pandoraPatRec:allOutcomes
 - Persist all possible reconstruction outcomes:
 - Clear cosmics
 - Slices reconstructed under both neutrino / TB and cosmic hypothesis

- Suggest the code first be released with the production of allOutcomes turned off
 - Users can turn on using a FHiCL switch
- The intention is for pandoraPatRec to remain the standard use-case
 - Users don't have to do anything differently!
- If allOutcomes has been switched on in the production of the files, users can then optionally decide to use the pandoraPatRec:allOutcomes collection to tailor / tune their own neutrino ID specific for their analysis
 - An ExternalEventBuilding module has been written to make this process as easy as possible, and includes an example neutrino ID tool. See next few slides!

-
- Consolidated pandora output is unchanged
 - Pandora is still only run once
 - Information about all reconstruction outcomes is persisted under separate instance name
 - No information is lost
 - User can have access to modify internal Pandora decisions downstream
 - External event building module allows users to run a custom neutrino ID tool
 - Technicalities are hidden behind a simple interface
 - PMT / CRT information can now be folded in!

Feature branches

- larpandora ([feature/larpandoracontent_v03_14_00](#))
 - Contains the bulk of C++ changes
 - XML updates. Some algorithms moved from master to worker instances to ensure identical event building internally and externally
- larpandoracontent ([feature/larpandoracontent_v03_14_00](#))
 - Small C++ change required to accomodate algorithm migration for protoDUNE use-case
 - A few minor changes to the output of protoDUNE events are expected
 - Suggest this be the release tagged as v03_14_00
- ubreco ([feature/larpandoracontent_v03_14_00](#))
 - XML updates as above
- dunetpc ([feature/larpandoracontent_v03_14_00](#))
 - XML updates as above

Advise move to Pandora external
v03_11_01 for TinyXML update

All branches built and tested for
MicroBooNE, protoDUNE, DUNE FD

Thanks Lorena, John & Steve

Pandora is an open project and new contributors would be extremely welcome.
We'd love to hear from you and we will always try to answer your questions!

Framework development

John Marshall (marshall@hep.phy.cam.ac.uk)
Mark Thomson (thomson@hep.phy.cam.ac.uk)

LArTPC algorithm development

John Marshall
Andy Blake (a.blake@lancaster.ac.uk)

MicroBooNE

Lorena Escudero (escudero@hep.phy.cam.ac.uk)
Joris Jan de Vries (jjd49@hep.phy.cam.ac.uk)
Jack Anthony (anthony@hep.phy.cam.ac.uk)
Andy Smith (asmith@hep.phy.cam.ac.uk)

protoDUNE

Steven Green (sg568@hep.phy.cam.ac.uk)
Stefano Vergani (sv408@cam.ac.uk)

Slack workspace

pandoraPFA.slack.com



UNIVERSITY OF
CAMBRIDGE

Lancaster
University





Backup

External event building

```
pandora_event_building:
{
  module_type: "LArPandoraExternalEventBuilding"

  NeutrinoIdTool:
  {
    tool_type: "SimpleNeutrinoId"
  }

  InputProducerLabel: "pandoraPatRec:allOutcomes"

  TrackProducerLabel: "pandoraAllOutcomesTrack"
  ShowerProducerLabel: "pandoraAllOutcomesShower"

  HitProducerLabel: "gaushit"
}
```

Event building module is configured to take a “class tool”, this is the how a user can include their own neutrino ID logic

This will be extended in the future to also work with the protoDUNE test-beam use-case. Only a small change.

By supplying a NeutrinoIdTool, the user is only exposed to the actual physics decision: Does this slice contain a neutrino?

The technicalities of working out which PFParticles are in which slice, ensuring no double counting, ... is abstracted!

Interface is LArSoft specific but could be generalised to include gallery uses if there is demand

- The neutrino ID tool passed to the external event building module must inherit from the abstract neutrino ID base class which has one pure virtual function:

```
virtual void ClassifySlices(SliceVector &lices, const art::Event &evt) = 0;
```

- The SliceVector contains Slice objects with member data:

```
float          m_nuScore;          ///< The neutrino score from Pandora  
PFParticleVector m_nuHypothesis;  ///< The slice as reconstructed under the neutrino hypothesis  
PFParticleVector m_crHypothesis;  ///< The slice as reconstructed under the CR hypothesis  
bool          m_isNeutrino;      ///< If the slice has been identified as a neutrino
```

- Users modify this flag using TagAsNeutrino() Or TagAsCosmic()
↑
- The external event building module will then output the correct reconstruction outcome for each slice. Users can't accidentally double count because the leg-work is hidden elsewhere!