Changes to radiological and supernova generators

Gleb Sinev

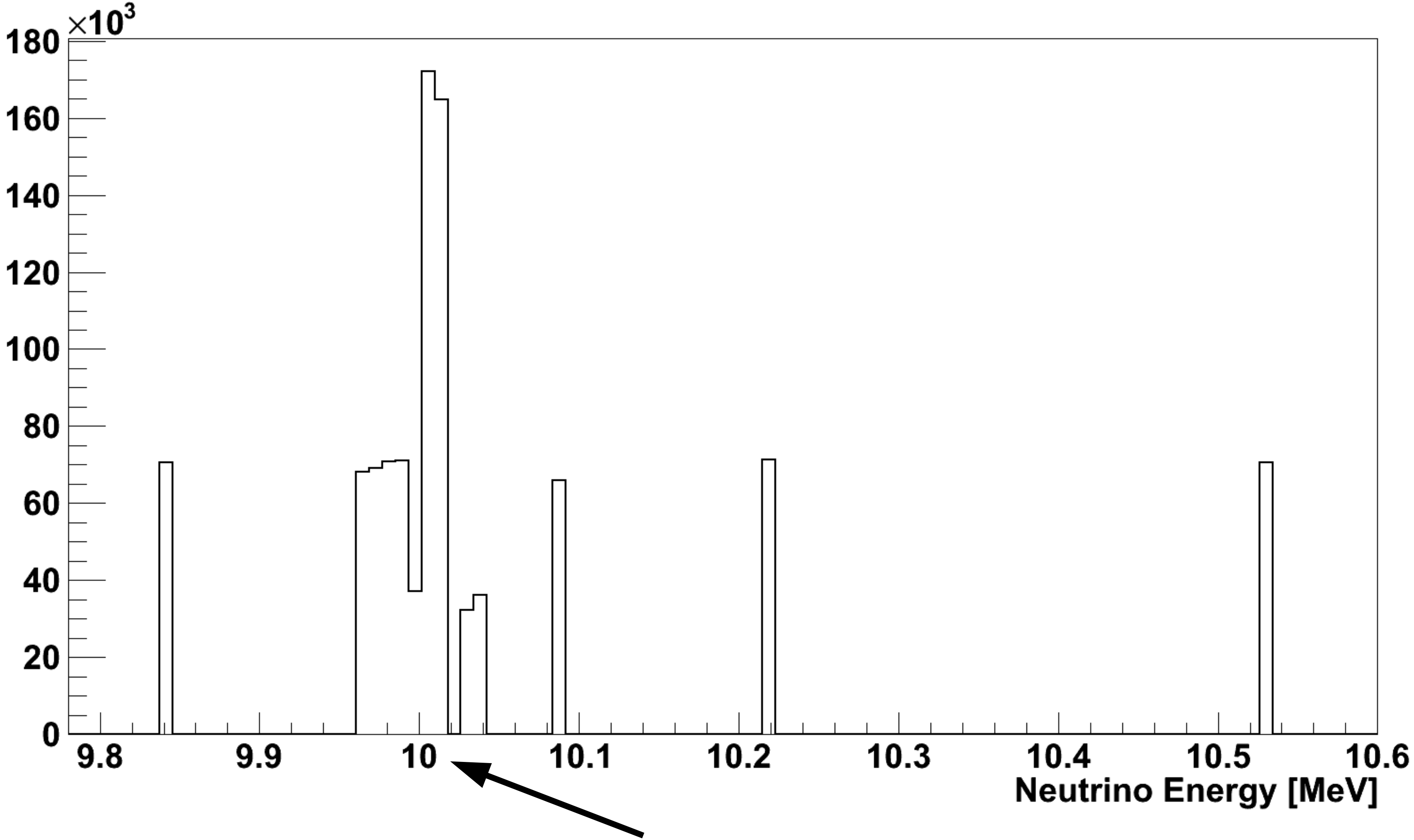
LArSoft Coordination Meeting

February 2, 2016

Supernova neutrino generator

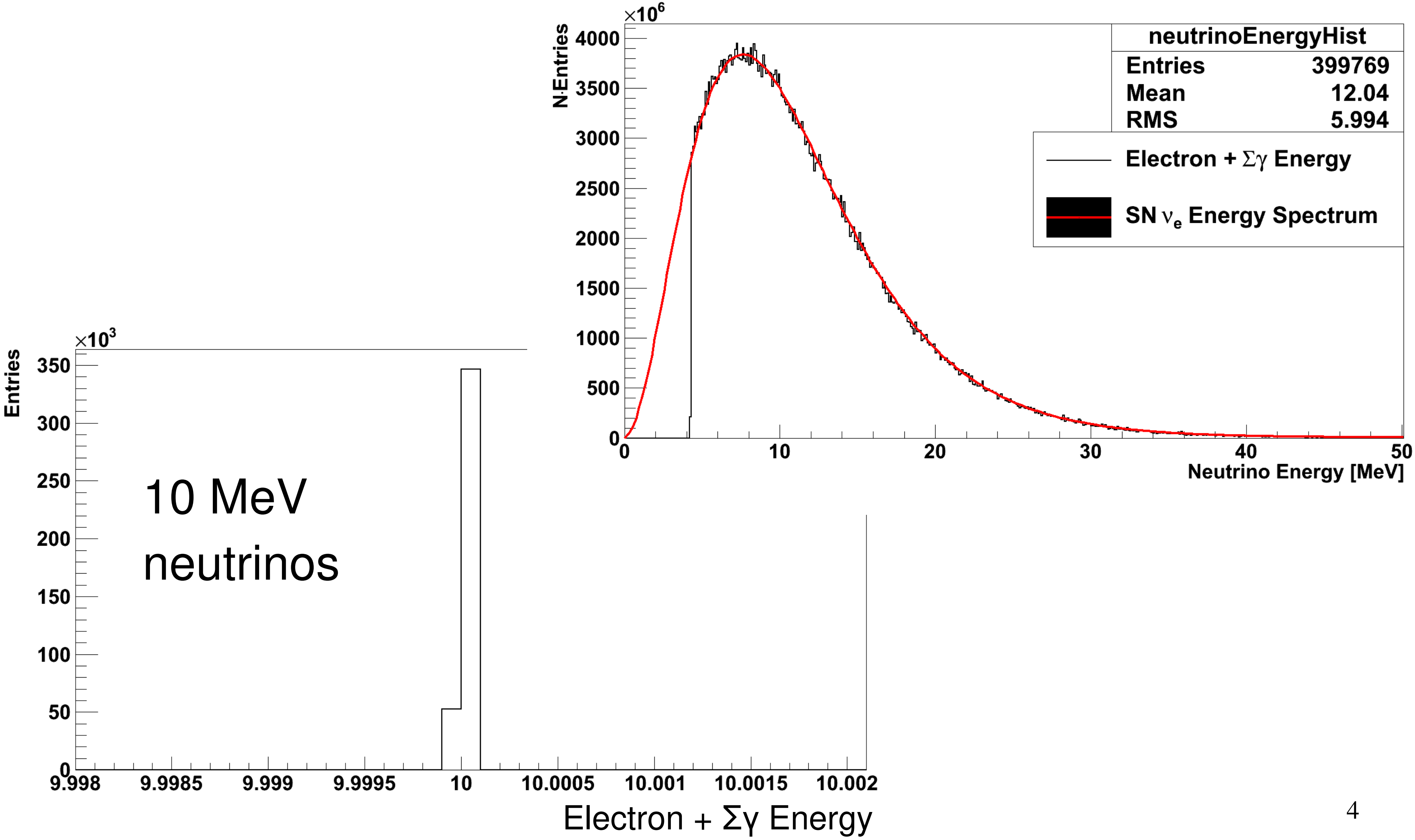
- A couple of months ago I wrote a module simulating ν_e-^{40}Ar CC interactions: `SNNueArCCGen_module`
- It contains a bug – there is a slight discrepancy between neutrino energy and energy of its products

Monoenergetic 10 MeV neutrinos



All events should have this energy

Bug is fixed



10 MeV neutrinos

Added new FhiCL parameters

```
EnergySpectrumFileName:“nue_spectrum.root”
MonoenergeticNeutrinos: false
NeutrinoEnergy:            10
MeanNumberOfNeutrinos:  40000
UsePoissonDistribution:  true
NumberOfNeutrinos:         10
NeutrinoTimeBegin:   -2246000.0
NeutrinoTimeEnd:      2246000.0
ActiveVolume0:      [ -363.376, -608.829,  -0.876 ]
ActiveVolume1:      [  363.376,  608.829, 463.904 ]
```

Changes to SN generator

- Feature branch in LArSim: `feature/gvsinev_SNGeneratorImprovements`
  - Fixed the bug
  - Can simulate monoenergetic neutrinos
  - Can simulate exactly N neutrinos

Radon generator

- Feature branch in LArSim: `feature/gvsinev_Rn222Background`
- Existing module `RadioGen_module`
- To simulate radon
  - Include RadioGen
  - `radiogen.Nuclide: [ “222Rn” ]`
  - `radiogen.BqPercc: [ 0.0141  ]`
- Produces single monoenergetic 3.5 MeV α-particles
- Thanks to Jason Stock and Juergen Reichenbacher for providing me with the generator

Added tree to OpFlashAna

- Feature branch in LArAna: `feature/gvsinev_PerEventFlashTree`
- Trees that exist have 1 flash per entry, which makes them hard to use if I calculate efficiency (and potentially slow if I have a lot of events)
- Previous trees containing flashes didn't have information about number of PEs on individual PDs
- A new tree with flash information stored for each event (off by default)
  - `MakePerEventFlashTree: false`

TTree branches

- Most of them are `std::vector< >`s

```
fPerEventFlashTree = tfs->make<TTree>("PerEventFlashTree","PerEventFlashTree");
fPerEventFlashTree->Branch("EventID",                     &fEventID,   "EventID/I");
fPerEventFlashTree->Branch("NFlashes",                    &fNFlashes,  "NFlashes/I");
fPerEventFlashTree->Branch("FlashIDVector",               &fFlashIDVector);
fPerEventFlashTree->Branch("YCenterVector",               &fYCenterVector);
fPerEventFlashTree->Branch("ZCenterVector",               &fZCenterVector);
fPerEventFlashTree->Branch("YWidthVector",                &fYWidthVector);
fPerEventFlashTree->Branch("ZWidthVector",                &fZWidthVector);
fPerEventFlashTree->Branch("FlashTimeVector",             &fFlashTimeVector);
fPerEventFlashTree->Branch("AbsTimeVector",               &fAbsTimeVector);
fPerEventFlashTree->Branch("FlashFrameVector",            &fFlashFrameVector);
fPerEventFlashTree->Branch("InBeamFrameVector",           &fInBeamFrameVector);
fPerEventFlashTree->Branch("OnBeamTimeVector",            &fOnBeamTimeVector);
fPerEventFlashTree->Branch("TotalPEVector",               &fTotalPEVector);
fPerEventFlashTree->Branch("NChannels",                   &fNChannels, "NChannels/I");
// The only way I can think of to record a two-dimensional variable-size array in a TTree
// is by flattening it into a one-dimension variable-size array
fPerEventFlashTree->Branch("PEsPerFlashPerChannelVector", &fPEsPerFlashPerChannelVector);
```

Summary

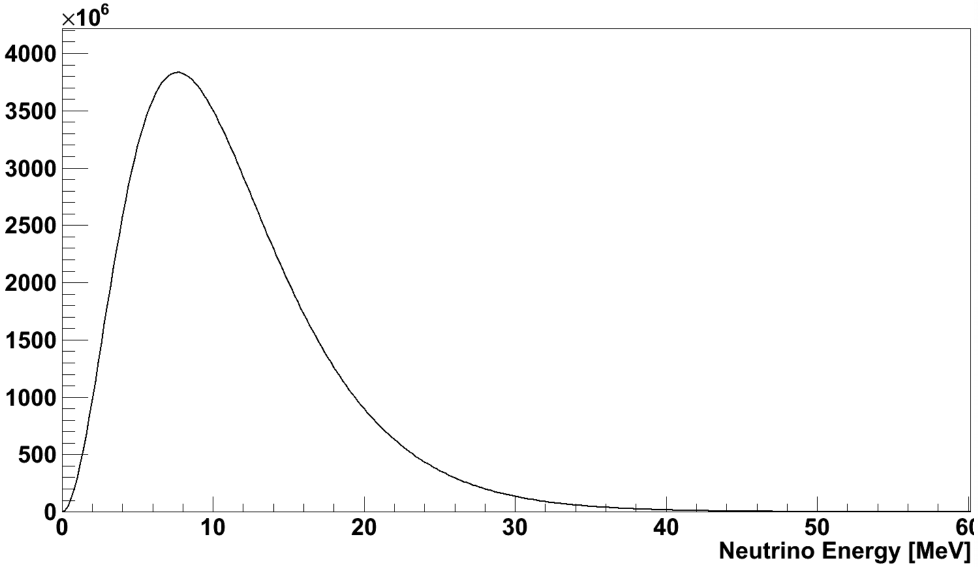
- These branches are updated to v04_35_00 and are ready to be merged
- LArSim
  - `feature/gvsinev_SNGeneratorImprovements`
  - `feature/gvsinev_Rn222Background`
- LArAna
  - `feature/gvsinev_PerEventFlashTree`

Backup slides

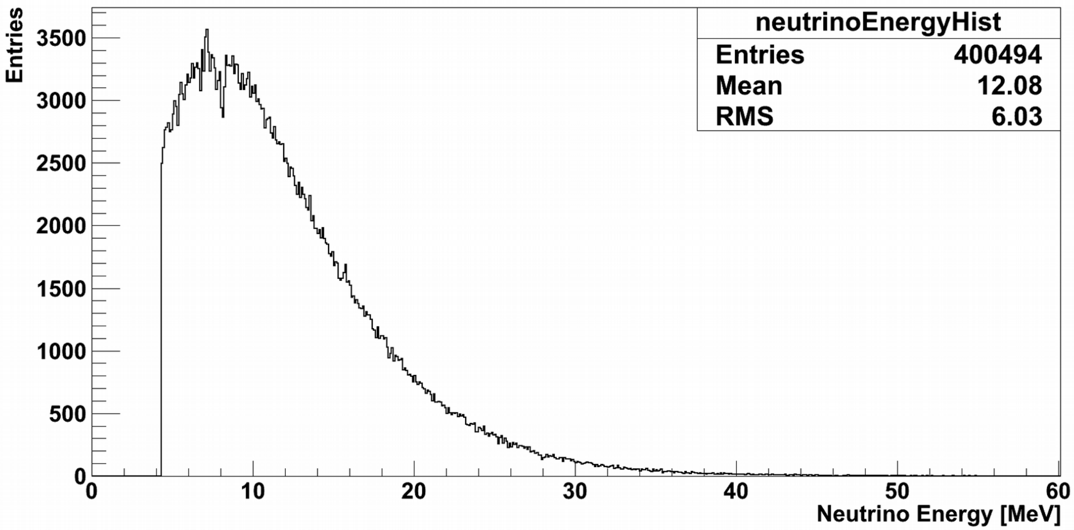
Energy bug in SN generator

- ν_e-^{40}Ar CC cross section and electron energy is calculated using a 21-level model
- Gammas are simulated using a 73-level model
- Levels of the first model are close to some levels of the second model
- Bug is fixed by calculating the electron energy using the 73-level model

Comparison of input and output



Primary neutrino energy



Electron energy
+ energy of all gammas