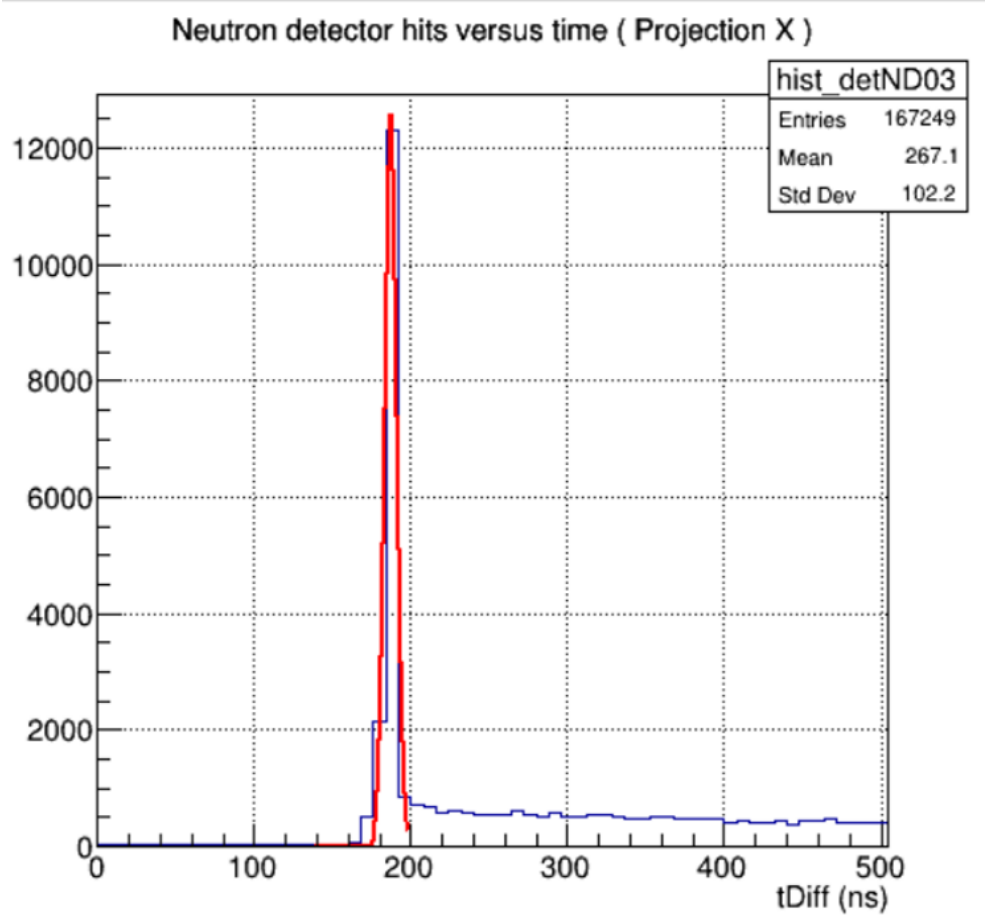
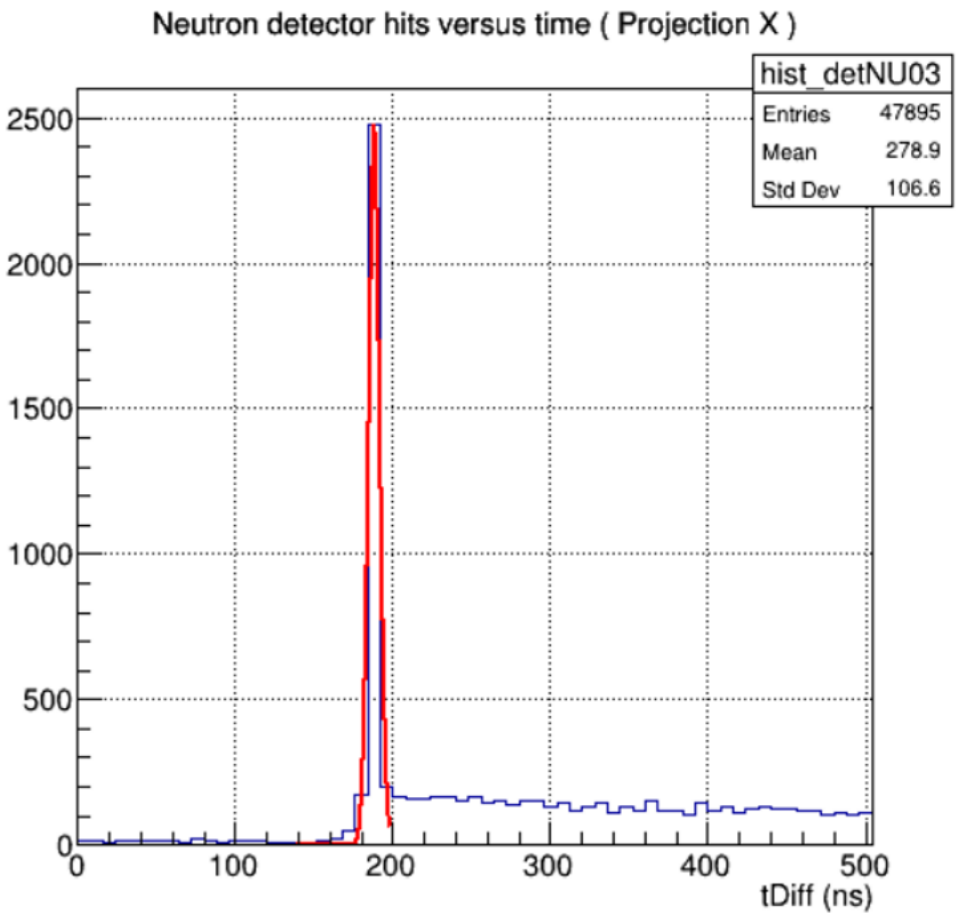
KU LEUVEN

Update muX meeting 09/12

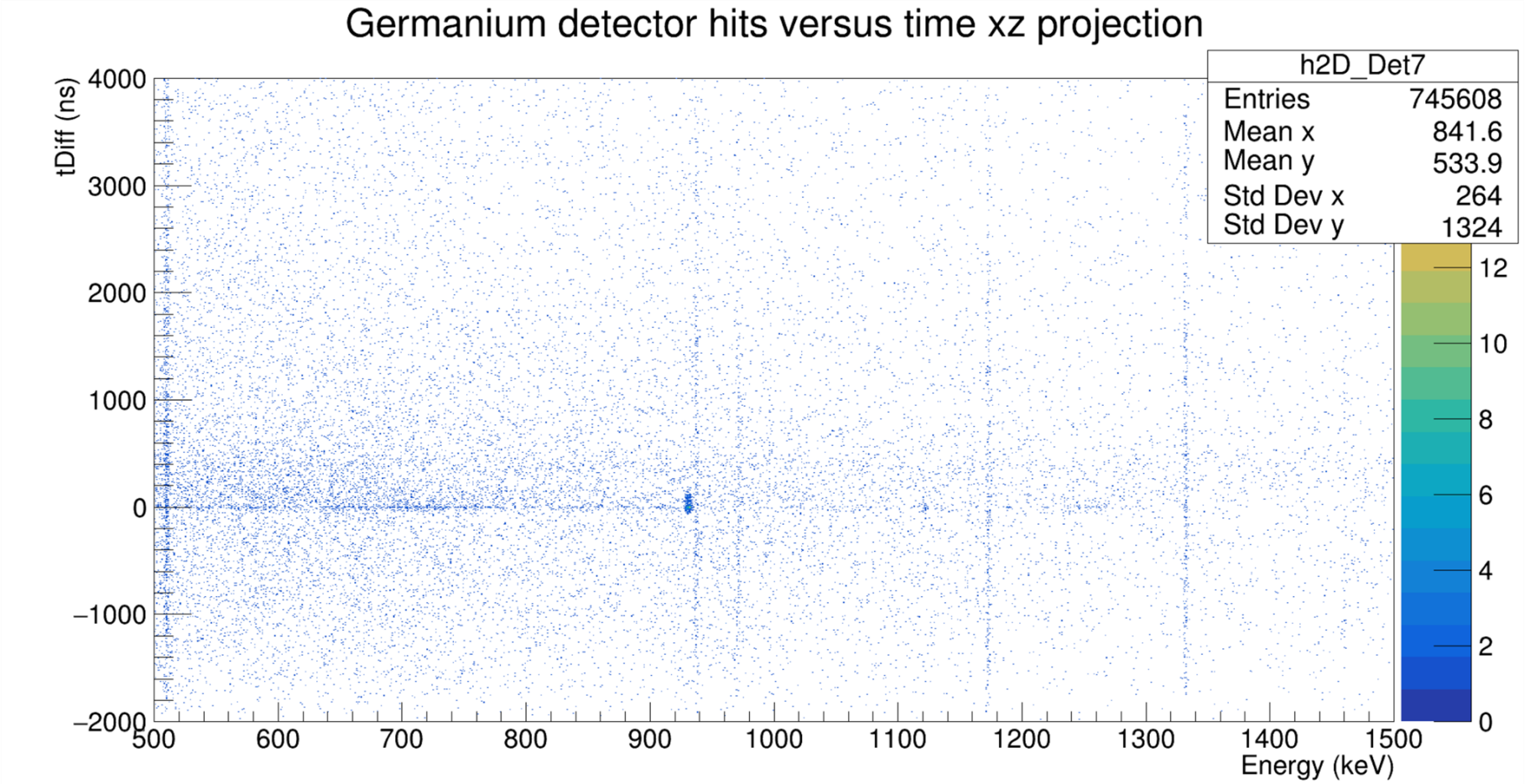
Michael Heines

Neutron time offsets

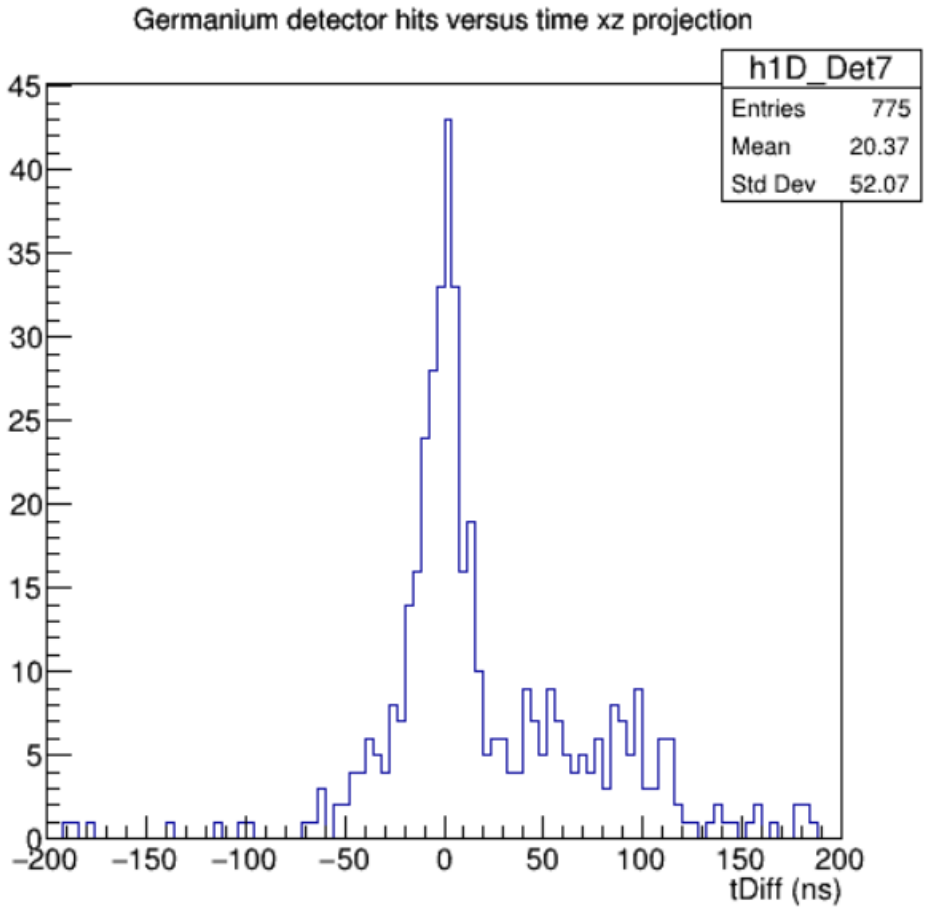


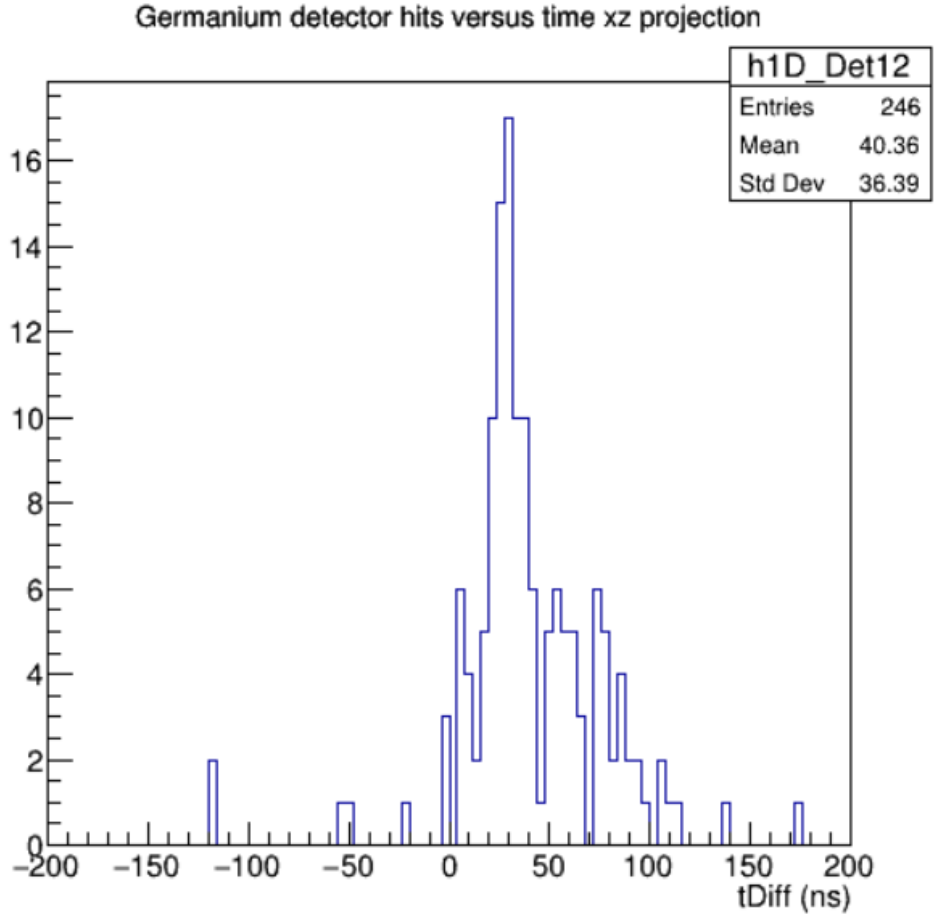


Delayed electron cut on BEGE's

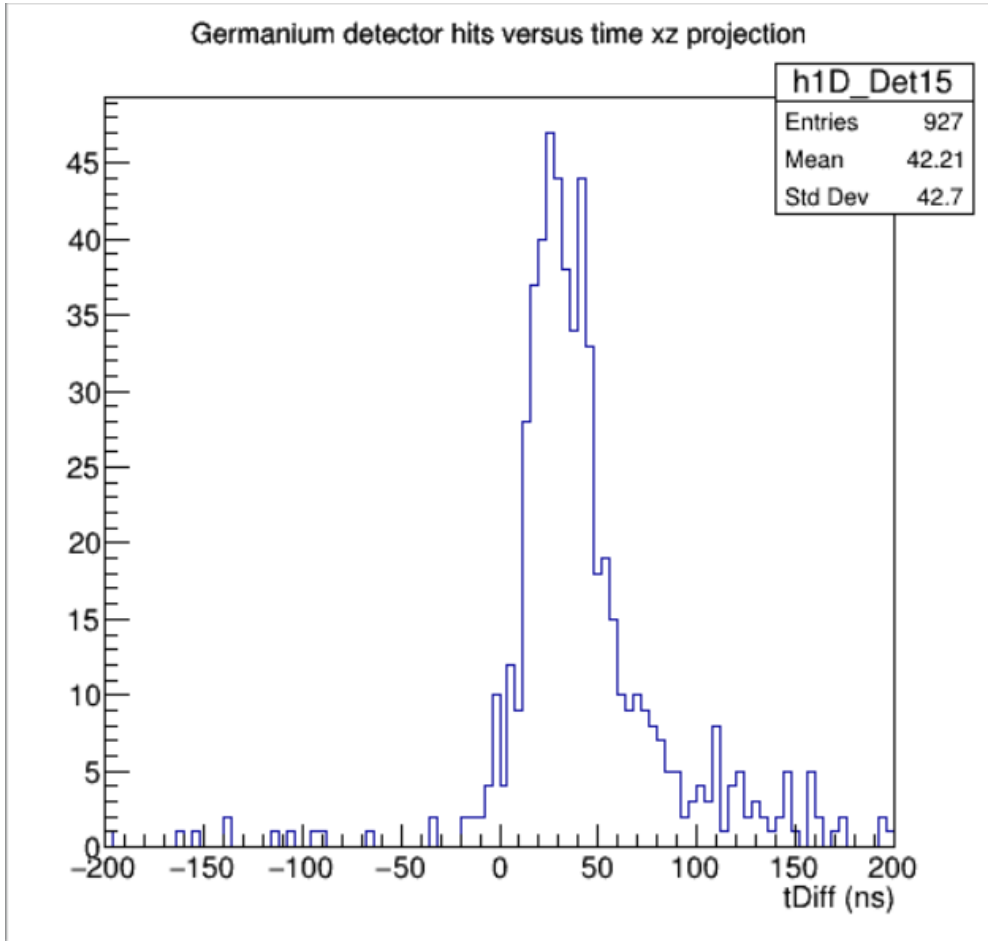


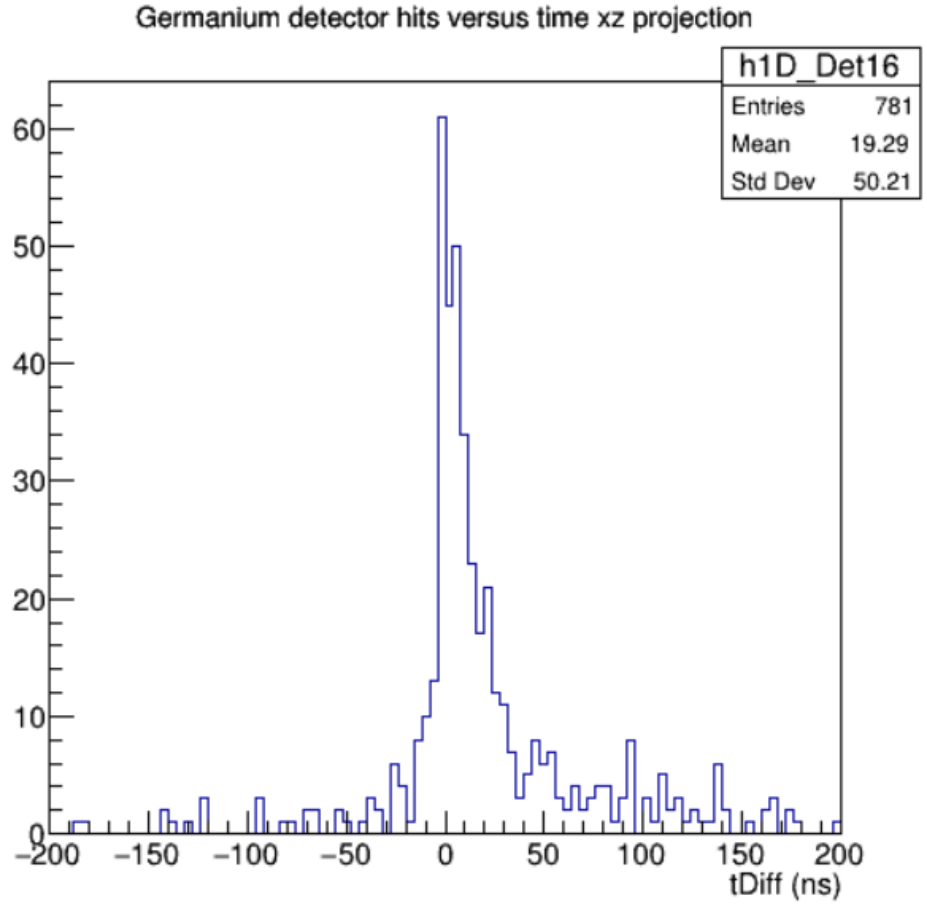
Energy cut around prompt (BEGE's)





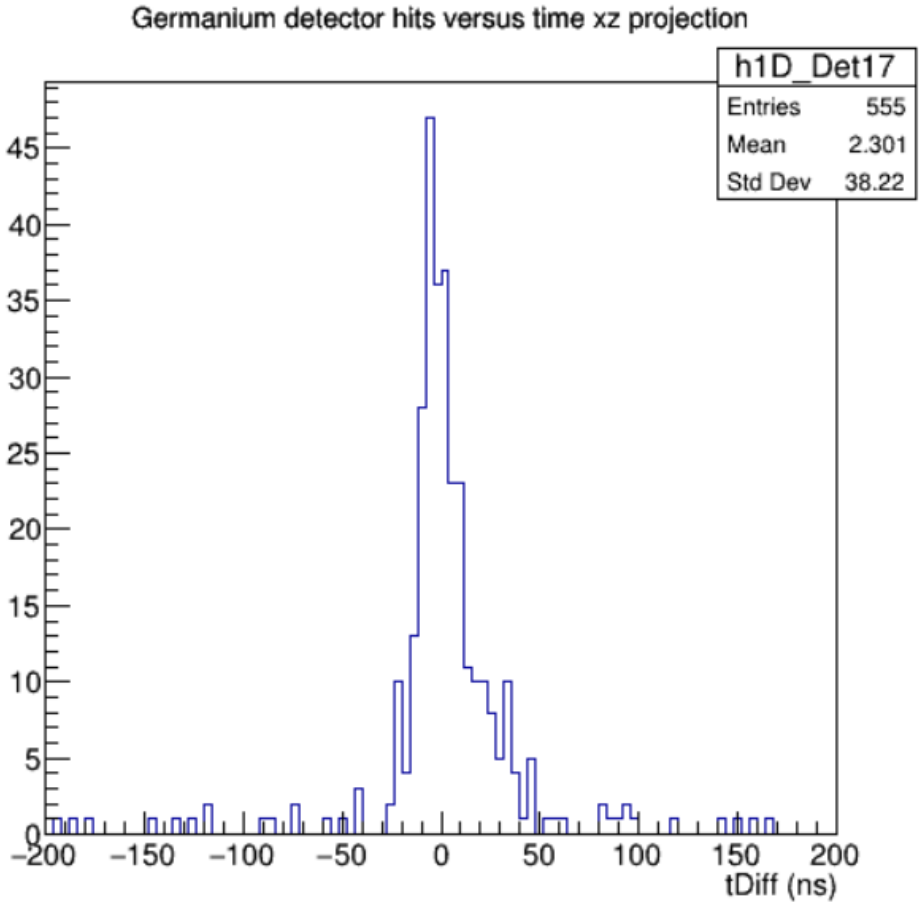
Energy cut around prompt (BEGE's)





Energy cut around prompt (BEGE's)

- Overall ok
  - Some minor late time tail (mainly for Ge04)
  - Typically 10-20 ns sigma
- Good enough?
- Other detectors: 5-10ns sigma



Writing to trees

- Filling the tree
  - Muon seed time
  - Vectors for others
  - Optionally save raw waveforms for neutron and germanium

- Basic version seems to work → Maybe push to the main branch next week?

```
//struct to map on flat TTree, vectors have to pointers
struct MuonEventForTree_t {
  ULong64_t muonTime;   // muon time in clock ticks
  std::vector<float>* geEnergies = new std::vector<float>();
  std::vector<float>* geEnergiesADC = new std::vector<float>();
  std::vector<float>* geTimes = new std::vector<float>();
  //std::vector<float>* geTriggerTimes = new std::vector<float>();
  std::vector<unsigned short int>* geChannels = new std::vector<unsigned short int>();
  std::vector<unsigned short int>* geModules = new std::vector<unsigned short int>();
  std::vector<unsigned short int>* geSamples = new std::vector<unsigned short int>();
  std::vector<float>* muonEnergies = new std::vector<float>();
  std::vector<float>* muonTimes = new std::vector<float>();
  std::vector<unsigned short int>* muonChannels = new std::vector<unsigned short int>();
  std::vector<unsigned short int>* muonModules =  new std::vector<unsigned short int>();
  std::vector<float>* electronEnergies = new std::vector<float>();
  std::vector<float>* electronTimes = new std::vector<float>();
  std::vector<unsigned short int>* electronChannels = new std::vector<unsigned short int>();
  std::vector<unsigned short int>* electronModules = new std::vector<unsigned short int>();
  std::vector<float>* neutronEnergies = new std::vector<float>();
  std::vector<float>* neutronTimes = new std::vector<float>();
  std::vector<unsigned short int>* neutronChannels = new std::vector<unsigned short int>();
  std::vector<unsigned short int>* neutronModules =  new std::vector<unsigned short int>();
  std::vector<unsigned short int>* neutronSamples = new std::vector<unsigned short int>();
};
```

Reading from trees

- Get MuonEventForTree_t from the tree
- Transform into MuonEvent_t
- Distribute from there
- Outside of main analyzer program

- Conversion function works
- Basic example/reading code under development

```
struct Hit_t {
  float energy; // energy, set to zero if you don`t want to save it. root will compress it
  float energyADC;
  float time; // time relative to the muon
  unsigned short int channel; // channel of struct module
  unsigned short int module;
  //trace
  std::vector<unsigned int> raw_samples;
  float trigger_time;
};

struct MuonEvent_t {
  ULong64_t muonTime;   // muon time in clock ticks
  std::vector<Hit_t> geHits;
  std::vector<Hit_t> muonHits;
  std::vector<Hit_t> electronHits;
  std::vector<Hit_t> neutronHits;
}
```

Conclusion

- Neutron time offsets are done
- If ELET ok → Fix time offsets for Ge to wrap up timing parameters
- Basic tree writing done? (except for waveform writing)
- Energy calibration: Fit from -1.5 sigma to 10 sigma → Seems to be working, but some minor optimization needed
  - For gold: Recalibrate about once per shift
  - For potassium: Recalibrate more often (~2 times/shift) → Look at stability and uncertainty on centroids

KU LEUVEN