

## dunetpc - Bug #22265

Task # 22198 (New): Address various issues in protodune-sp reconstruction

### Possible memory leaks in GausHitFinder

04/02/2019 09:43 AM - Tingjun Yang

<b>Status:</b>	Closed	<b>Start date:</b>	04/02/2019
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Description</b>			
Valgrind showed the following warning:			
<pre>==22281== 16 bytes in 4 blocks are definitely lost in loss record 2,151 of 63,444 ==22281==    at 0x4C2A243: operator new(unsigned long) (vg_replace_malloc.c:334) ==22281==    by 0xD6DE25E: TFormula::HandleParamRanges(TString&amp;) (TFormula.cxx:1176) ==22281==    by 0xD6E093C: TFormula::PreProcessFormula(TString&amp;) (TFormula.cxx:1644) ==22281==    by 0xD6D99C1: TFormula::TFormula(char const*, char const*, bool, bool) (TFormula.cxx: 403) ==22281==    by 0xD6A6354: TF1::TF1(char const*, char const*, double, double, TF1::EAddToList, boo l) (TF1.cxx:566) ==22281==    by 0x61FD2DEA: reco_tool::BaselinedGausFitCache::CreateFunction(unsigned long) const (PeakFitterGaussian_tool.cc:51) ==22281==    by 0x2828FC1C: hit::GausFitCache::Get(unsigned long) (GausFitCache.cxx:49) ==22281==    by 0x61FD051F: reco_tool::PeakFitterGaussian::findPeakParameters(std::vector&lt;float, s td::allocator&lt;float&gt; &gt; const&amp;, std::vector&lt;reco_tool::ICandidateHitFinder::HitCandidate, std::allo cator&lt;reco_tool::ICandidateHitFinder::HitCandidate&gt; &gt; const&amp;, std::vector&lt;reco_tool::IPeakFitter:: PeakFitParams, std::allocator&lt;reco_tool::IPeakFitter::PeakFitParams&gt; &gt;&amp;, double&amp;, int&amp;) const (Pea kFitterGaussian_tool.cc:205) ==22281==    by 0x3521CAC8: hit::GausHitFinder::produce(art::Event&amp;) (GausHitFinder_module.cc:380) ==22281==    by 0x66C0BE0: art::EDProducer::produceWithFrame(art::Event&amp;, art::ProcessingFrame con st&amp;) (EDProducer.cc:91) ==22281==    by 0x6797845: art::detail::Producer::doEvent(art::EventPrincipal&amp;, art::ModuleContext const&amp;, std::atomic&lt;unsigned long&gt;&amp;, std::atomic&lt;unsigned long&gt;&amp;, std::atomic&lt;unsigned long&gt;&amp;) (P roducer.cc:125) ==22281==    by 0x275B7D37: art::WorkerT&lt;art::EDProducer&gt;::implDoProcess(art::EventPrincipal&amp;, art ::ModuleContext const&amp;) (WorkerT.h:198) ==22281==</pre>			

### History

#### #1 - 04/02/2019 01:09 PM - Gianluca Petrillo

The object `reco_tool::BaselinedGausFitCache` is, you guess, a cache. It is derived from `reco_tool::GausFitCache` which turns out to explicitly state that its TF1 functions [are never deleted](#). I probably chose that to avoid the ROOT ownership trouble (what if ROOT decides afterwards to delete the function? what if it still remembers it?).

More to the point: that is a design choice. It may be changed, but I am afraid it would take a good amount of ROOT-guess-work at the risk of *occasional* crashes at the end of a job<sup>1</sup>.

Is the leak itself a problem? I think it shouldn't, in that the memory would be released at the end of the job anyway. Or is the annoyance the fact that it appears in valgrind output? If so, how do you think the annoyance could be eased?

<sup>1</sup> This is good material for nightmares.

#### #2 - 04/02/2019 01:18 PM - Tingjun Yang

Hi Gianluca,

Thanks for the detailed information. As long as it is not a real memory leak, I don't think we need to change anything. I can live with the valgrind warning.

Thanks,  
Tingjun

**#3 - 04/02/2019 01:19 PM - Tingjun Yang**

- % Done changed from 0 to 100

- Status changed from New to Resolved

**#4 - 05/11/2019 03:31 PM - Tingjun Yang**

- Status changed from Resolved to Closed