

Validating Decay0

Pierre Lasorak

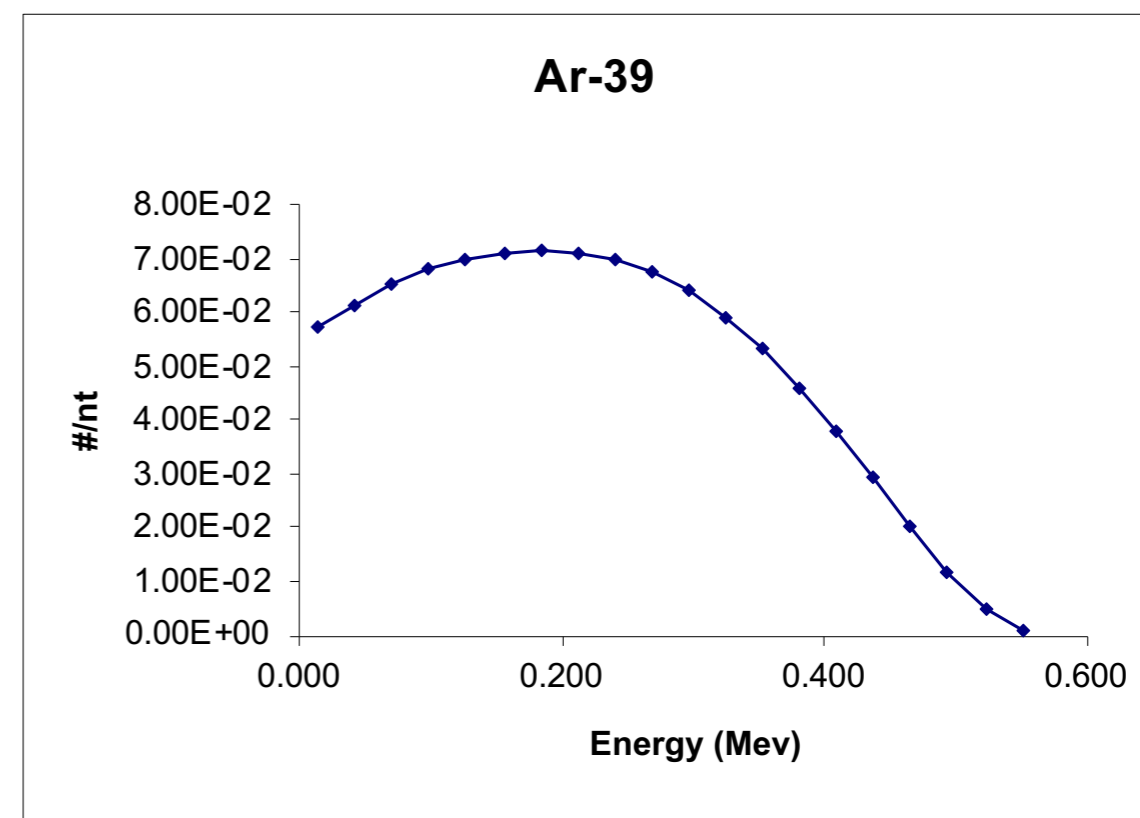
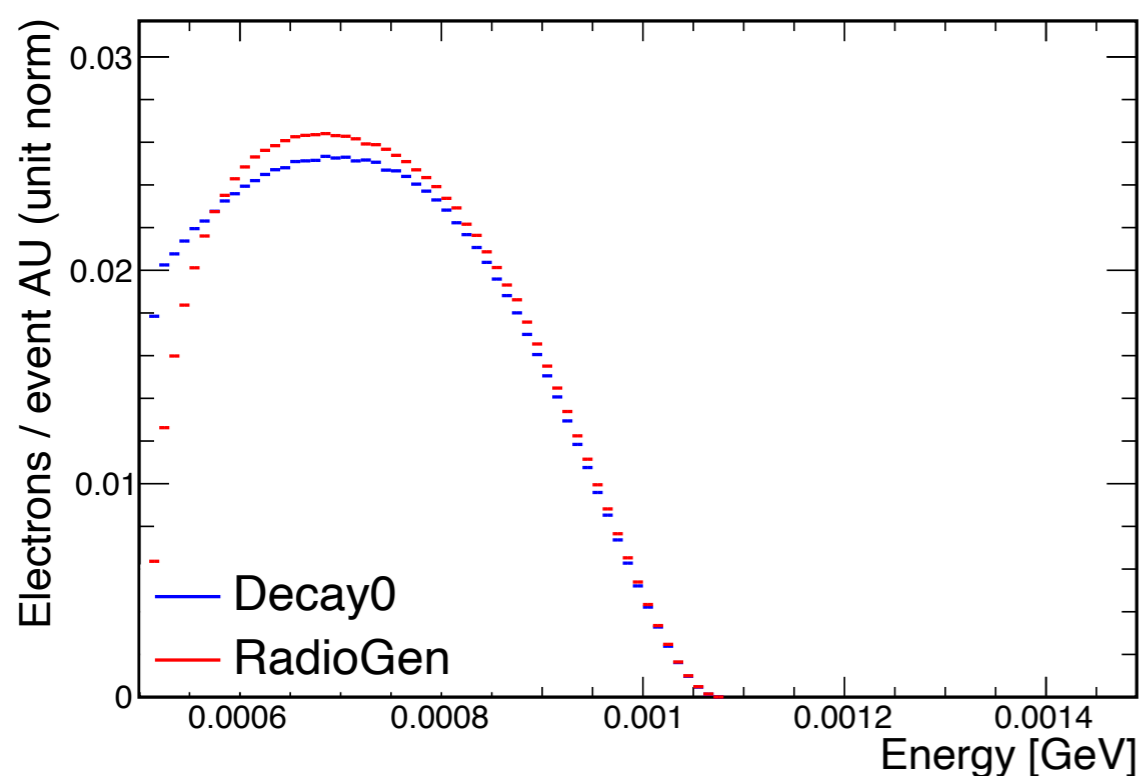
- Last year, I implemented an interface to BxDecay0 in LArSoft.
 - BxDecay0: <https://github.com/BxCppDev/bxdecay0>
 - Interface lives in “larsimrad” (for LArSoft radiation simulation):
 - https://github.com/LArSoft/larsimrad/blob/develop/larsimrad/BxDecay0/Decay0Gen_module.cc
- We still haven't checked that it is doing the correct thing.

- No change - we can keep the RadioGen_module:
 - Argon 39: beta decays in the LAr
 - Krypton 85: beta decays in the LAr
 - Cobalt 65: beta decays on the APA
 - Potassium 40: beta decays on the CPA
- Change - we need Decay0_module, or to do some recalculations:
 - Argon 42: beta decays in the argon + subsequent decay of the K42 (20% in the LAr, 80% on the Cathode)
 - Uranium 238: full decay chain including Bismuth/Polonium
 - Rock neutrons: new rate of $\sim 10\text{Hz}$ (c.f. Aran's updates)
- New:
 - Potentially many things:
<https://indico.fnal.gov/event/47792/contributions/208383/attachments/139932/175769/FangBGTFMeeting10Feb2021.pdf>

- The new Decay0_module features:
 - Uses BxDecay for beta/alpha decays
 - Potentially different spectra, and potentially new decays
 - Ability to directly generate in a specified geometry volume and material, and a new method to calculate the rate
 - Potentially different rate and position for the decays
- Essentially this means that I need to verify for each channel:
 - The position of the decays
 - The rate of the decays
 - The particle content of the decays
 - The spectra of the decays

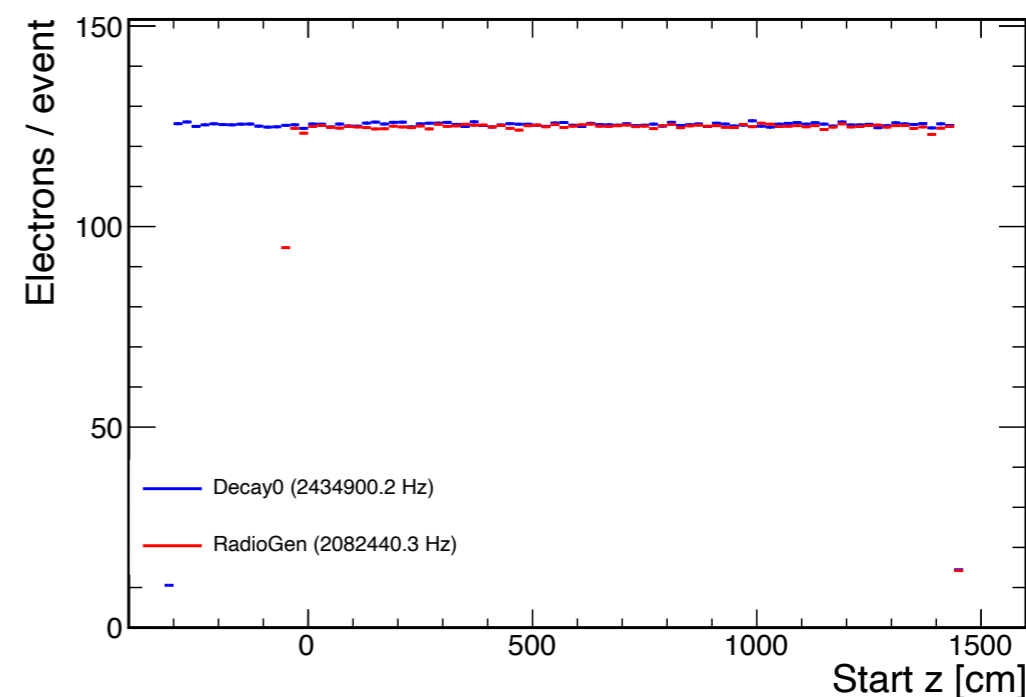
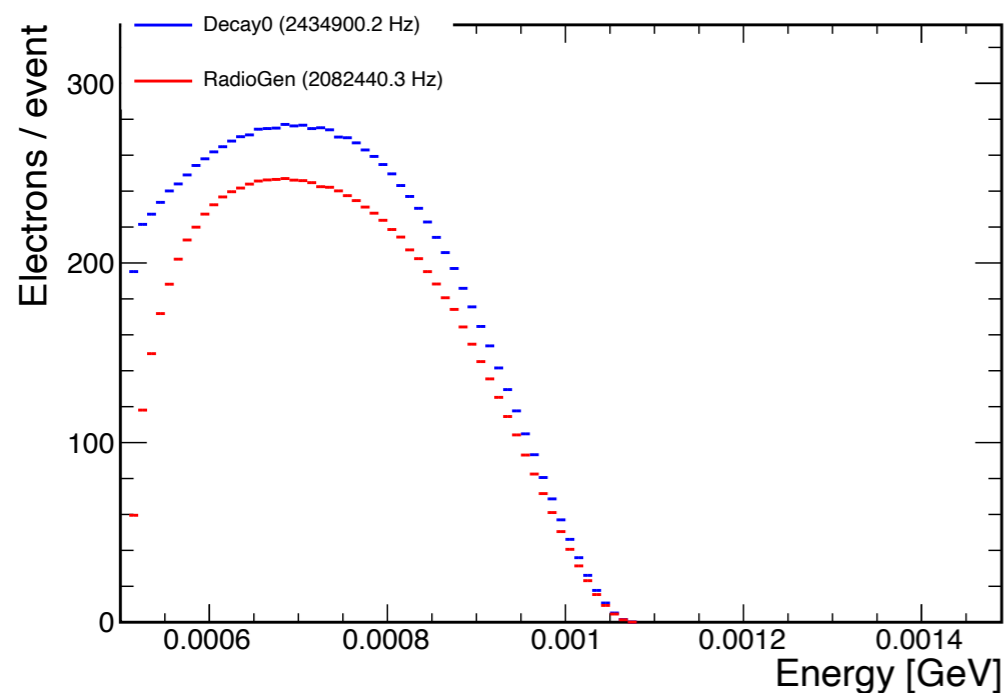
- Following are what happens if you run Decay0 with the most naïve reimplementations of the radiological model.
 - This is all run on the DUNE 1x2x6 geometry
 - I haven't tried to recalculate rates etc.

- Unchanged channel, we could still use RadioGen_module if we wanted to.
- If I remember well, SDSMT calculated many of the spectra.
 - I'm not trying to point fingers here, many of the differences that we are seeing probably don't make any difference
 - Found it interesting to compare with another calculation: <https://www.doseinfo-radar.com/RADARDecay.html>

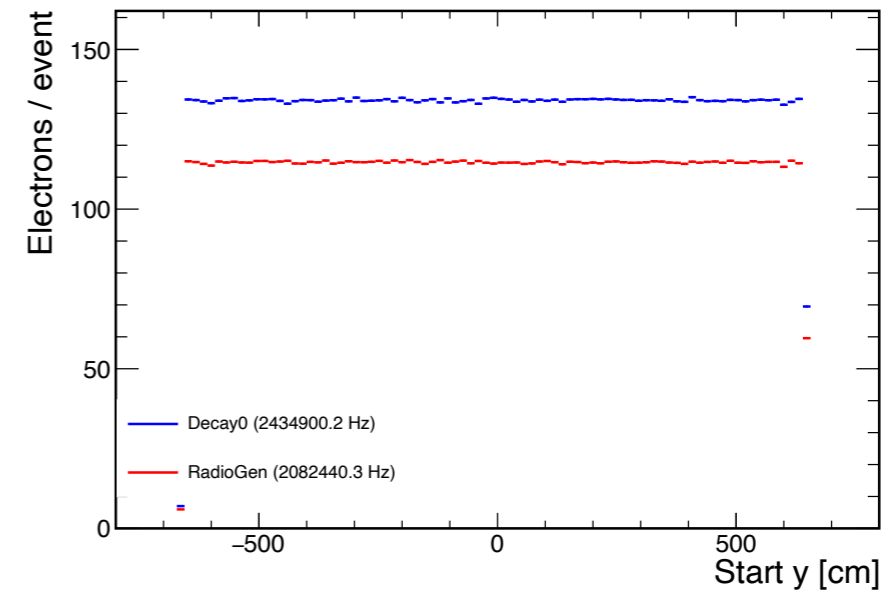
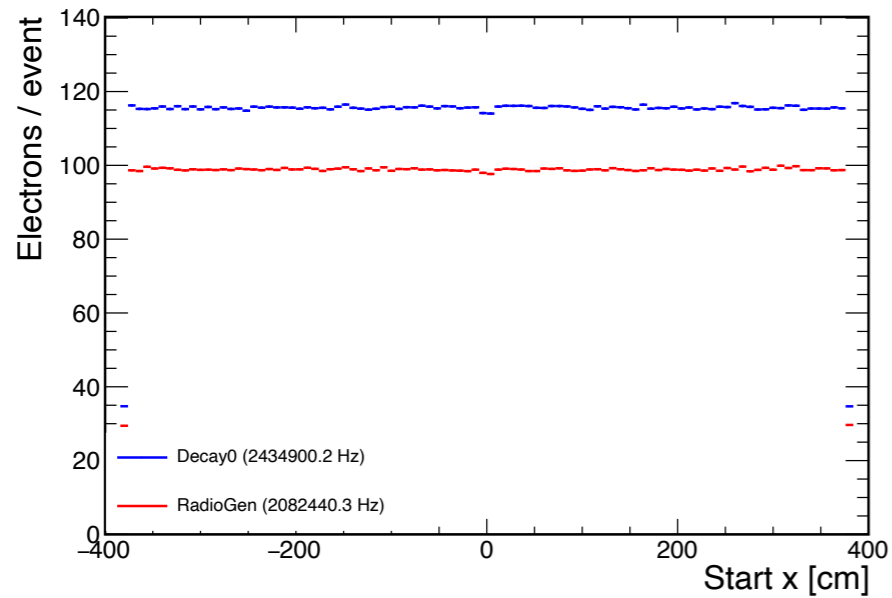


Doseinfo-radar website

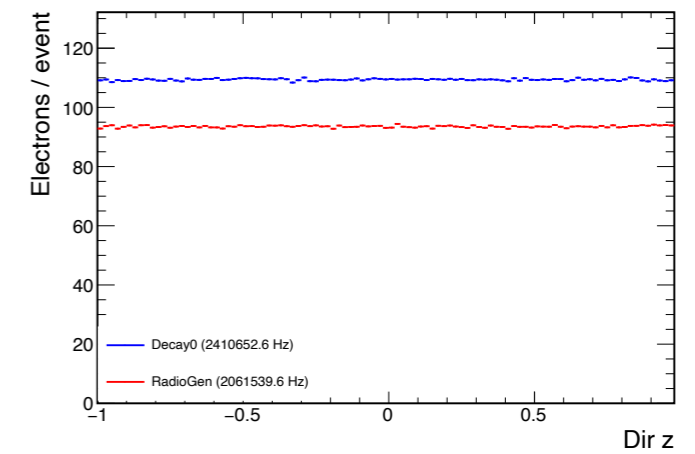
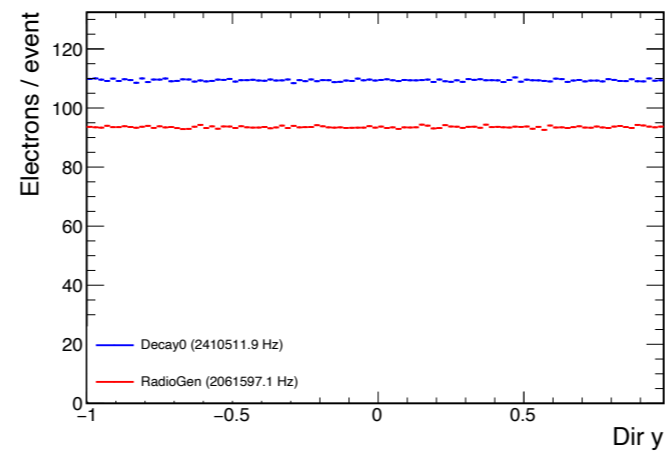
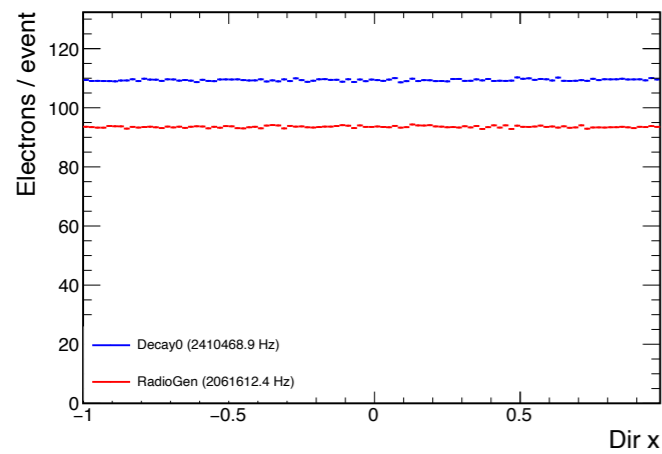
- Comparing normalised spectra
- Fairly big difference explained by the extra decays generated in the non instrumented part of the LAr



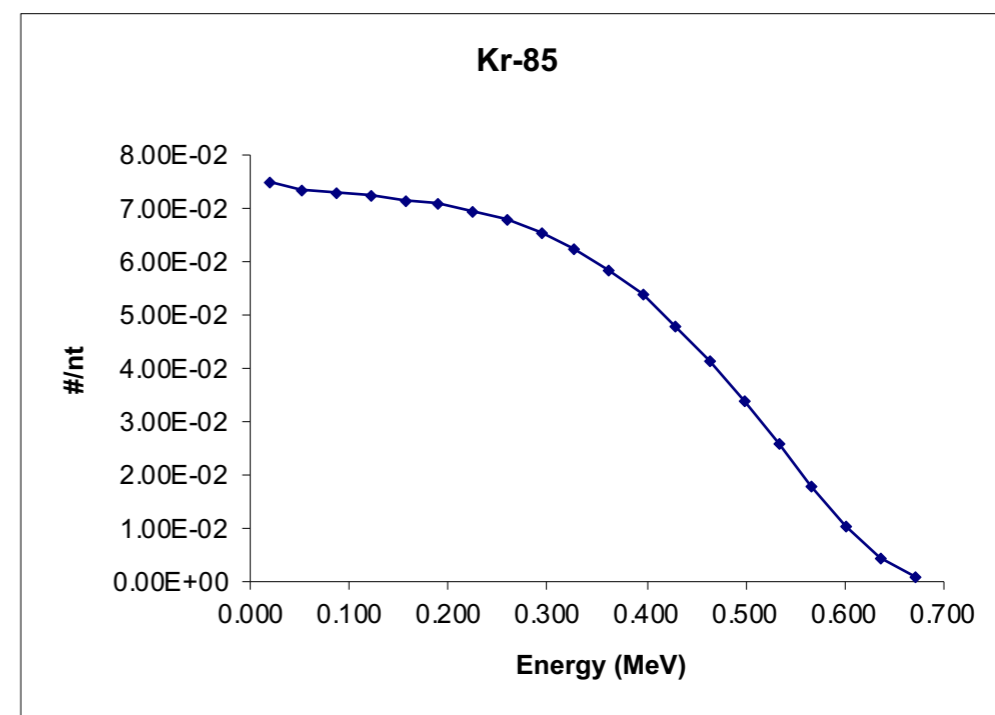
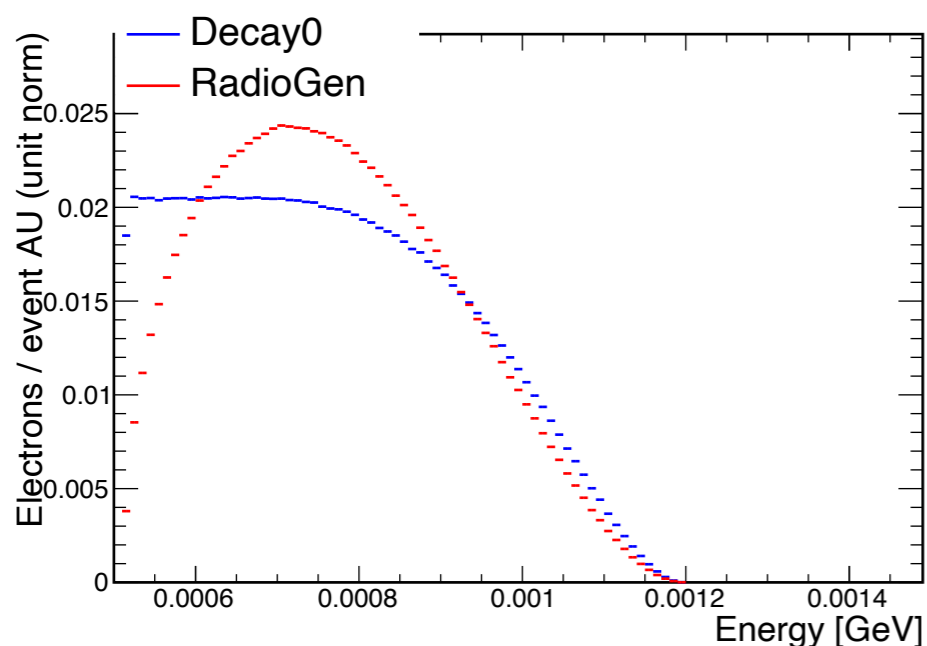
- Rest looks good, x and y positions:



- Electron direction:

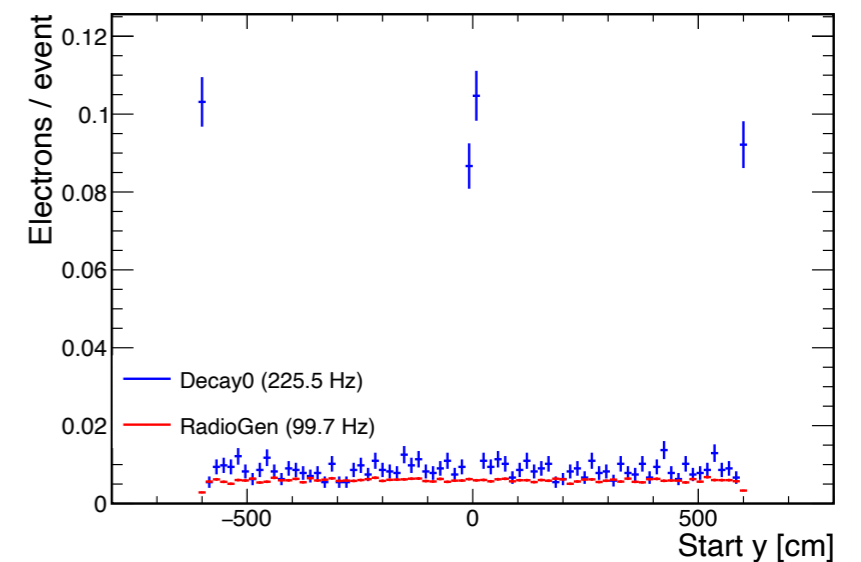
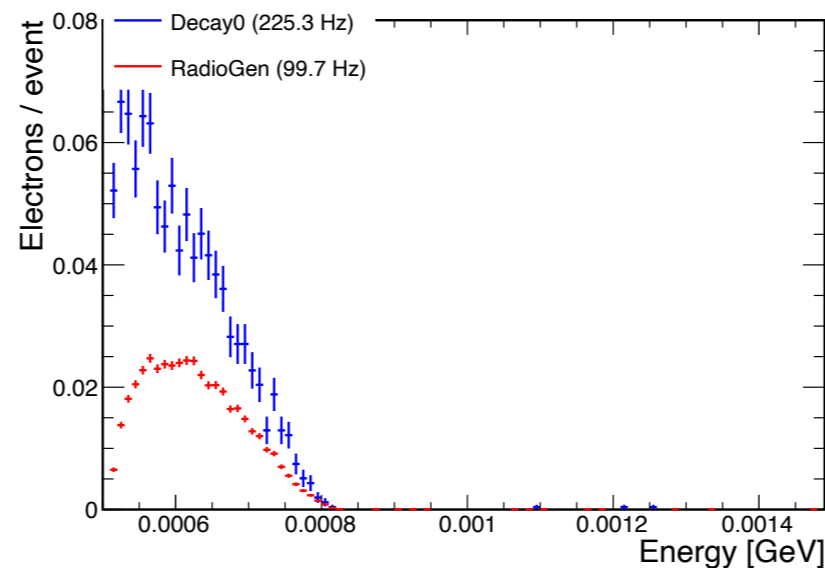
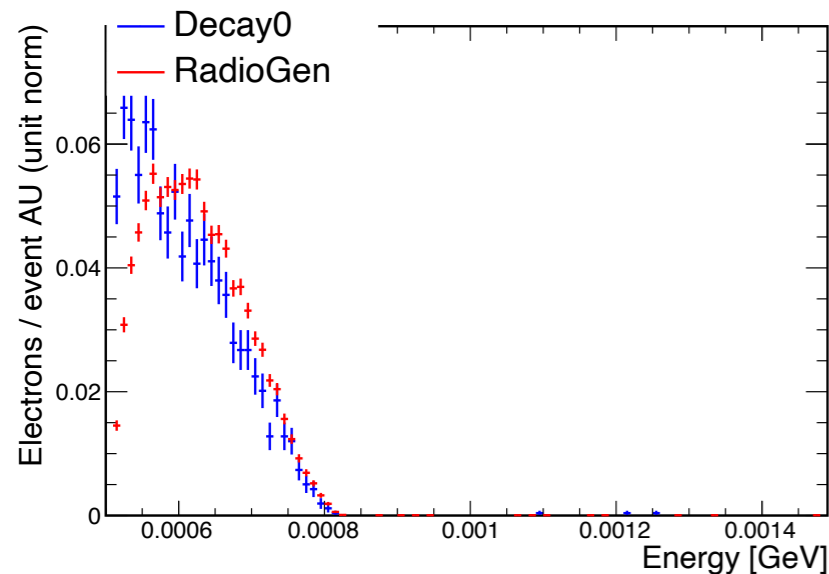


- Again we don't have to change this one
- Quite different spectra, but overall very low energy...
- Same behaviour for the rate as Argon 39. These decays are generated everywhere in the LAr with Decay0, and not with RadioGen.

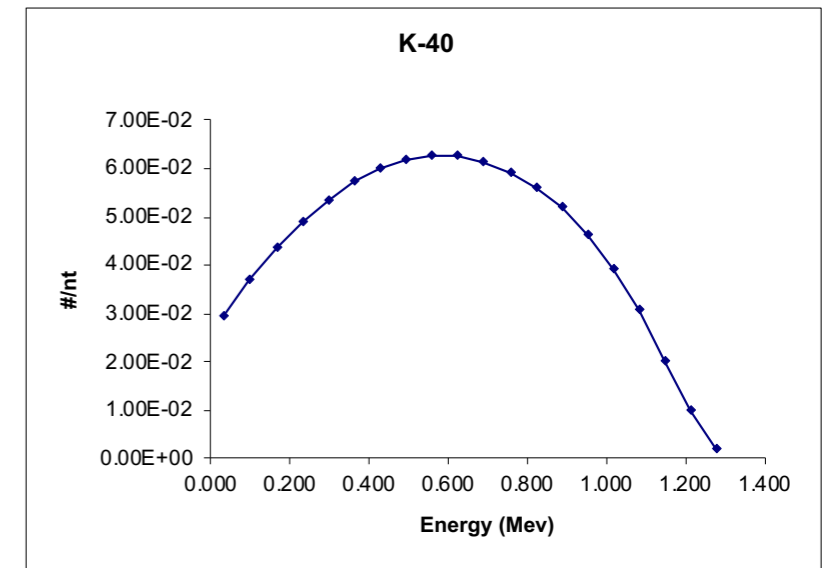


Doseinfo-radar website

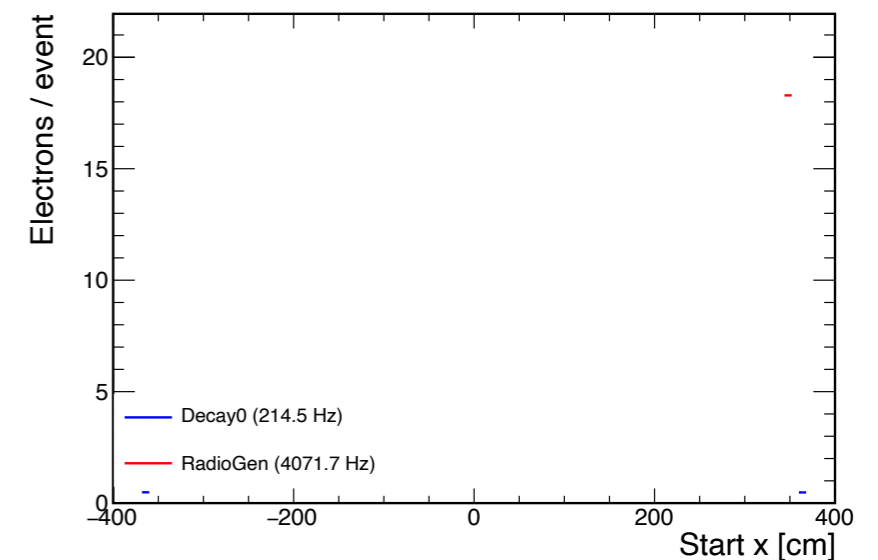
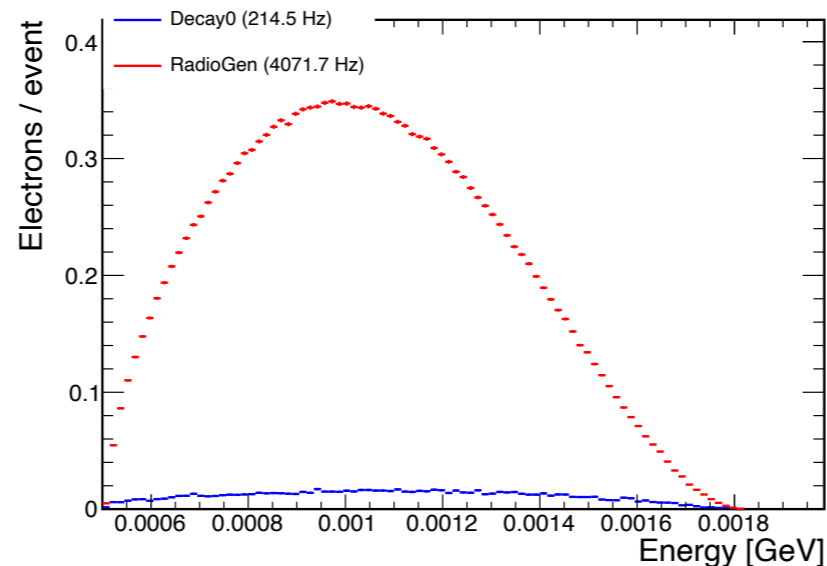
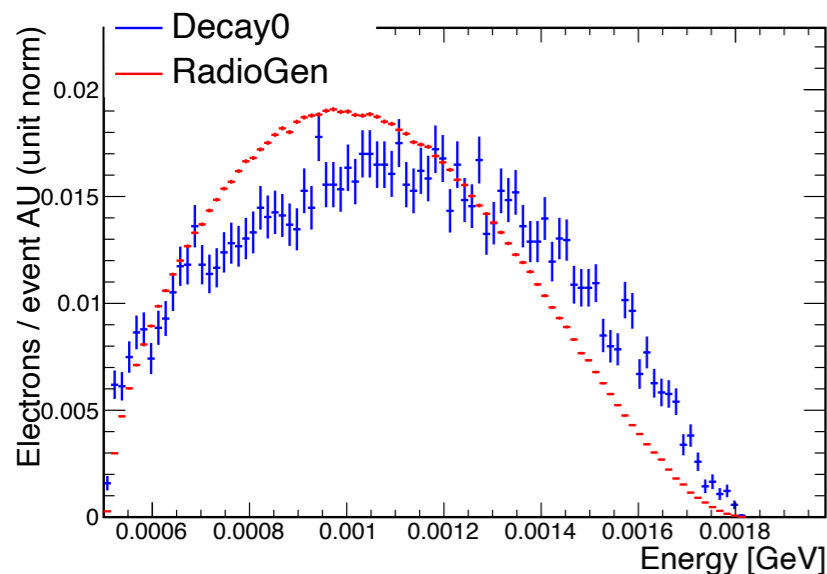
- Again, we don't need to use Decay0 for this one
- Spectra looks consistent
 - Not able to compare with the Doseinfo-radar website, as this channel isn't here
- The rate however is completely wrong, this seem to be due to a couple of extra places where we get more decays than expected, I need to check that in more details.



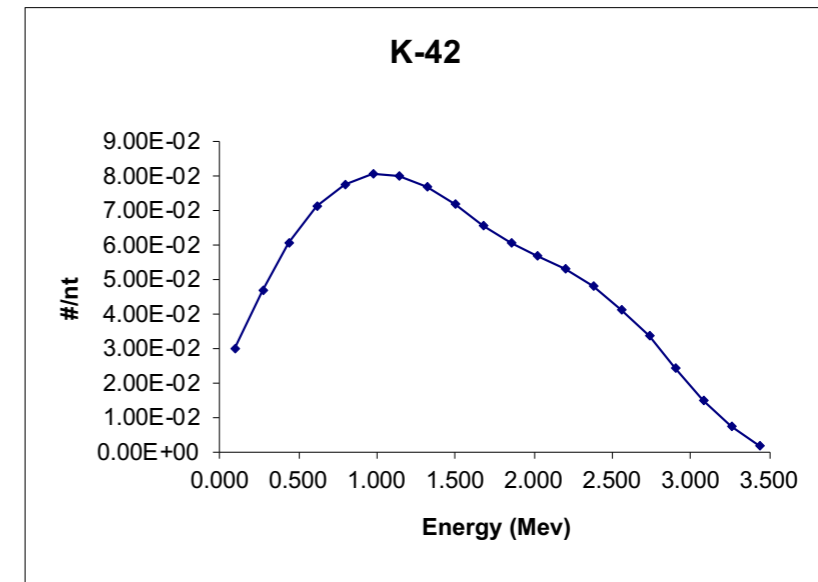
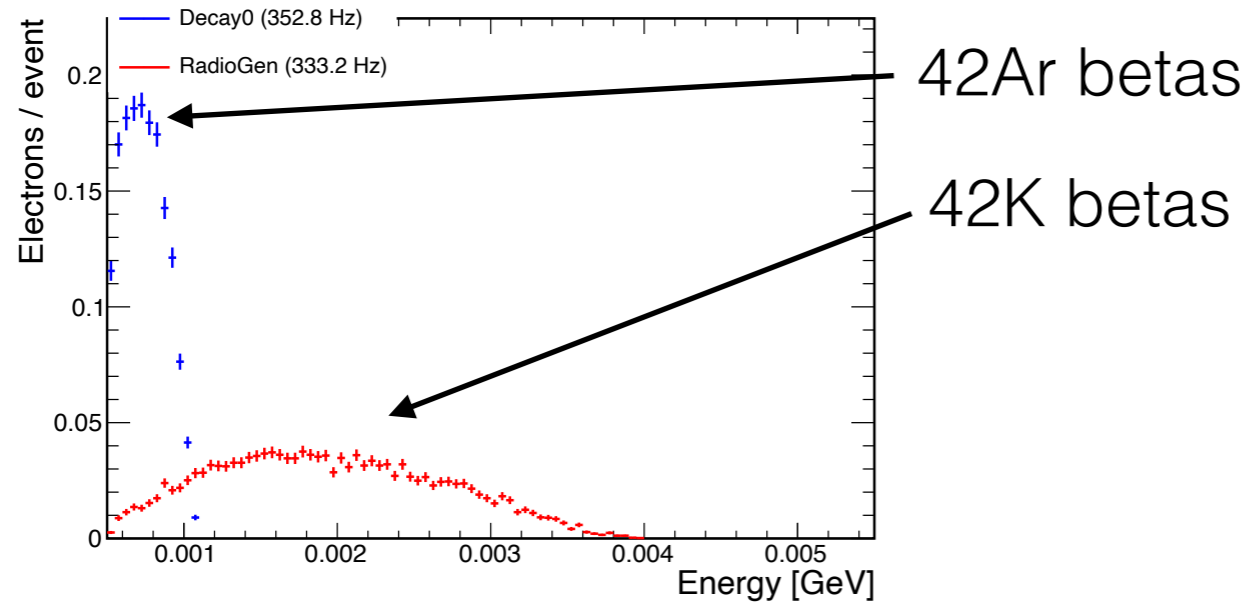
- Again, we wouldn't need to use Decay0 for this one
- Spectra looks consistent
- Again the rate is completely wrong,
 - Decay0 seem to generate on both the CPAs, whereas RadioGen isn't.



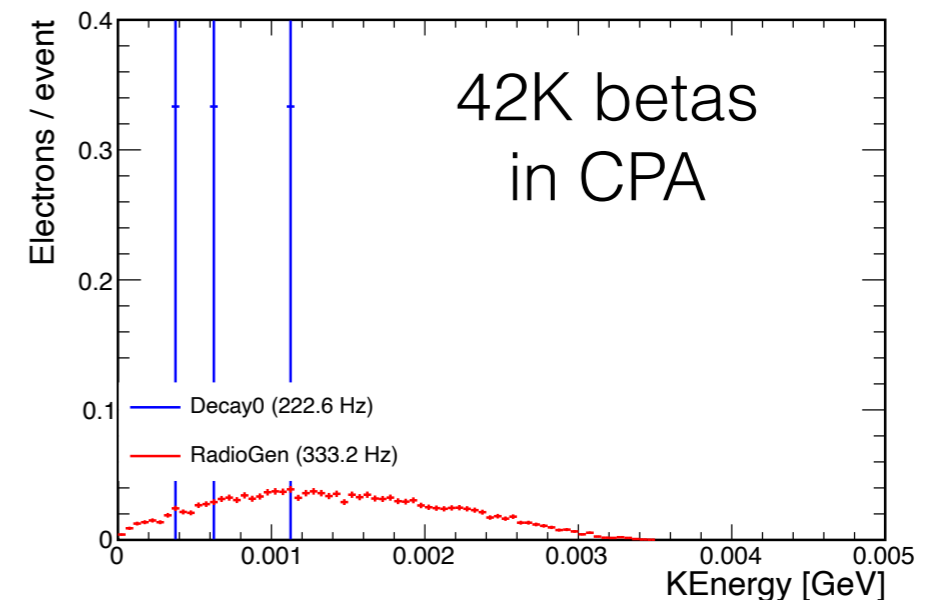
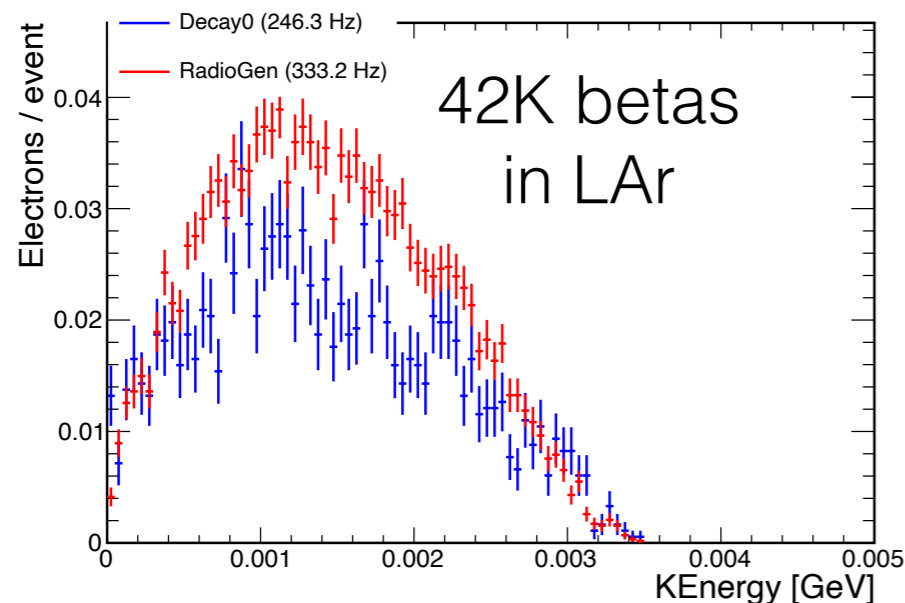
Doseinfo-radar website



- For this, we want to get both the initial Argon 42 beta decay and the late Potassium 42 decay
 - 80% go to CPA
 - 20% stay in LAs
- RadioGen only simulates the Potassium 42.
- Again Decay0 rates are complete gibberish



Doseinfo-radar website



- Initial, naïve validations of Decay0 are looking good, but there is still quite a bit of work to start using Decay0 in dunetpc
 - All the spectra checked are looking consistent
 - Still need to understand the issue with the rate for the APA and the CPA decays and for the Argon 42.
 - Probably will need some modifications to larsimrad in the near future.
- Working towards implementing correctly the U238 chain, which is what we actually want.