



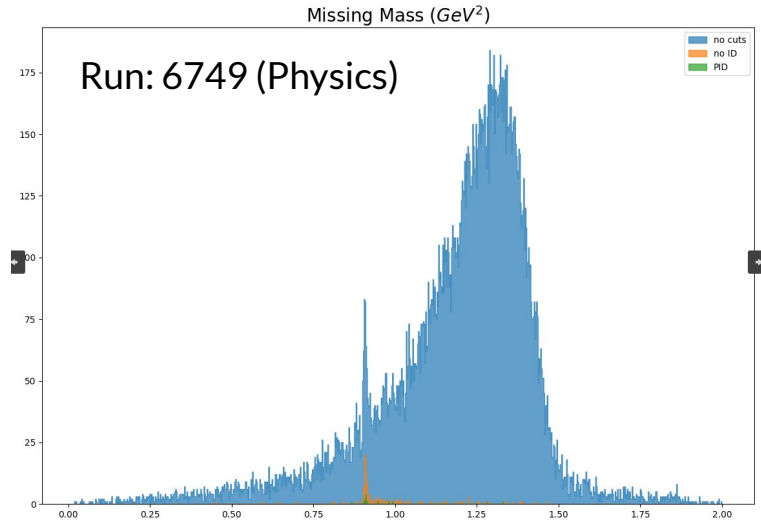
# Kaon LT Status Update

April 15th, 2020

Richard Trotta

# New PID script

- Like the lumi script, old C++/ROOT based script was giving lots of issues with newer ROOT files
  - Converted it over to python just like the lumi script
- There are output plots and a csv, like the lumi script



# Restructured UTIL\_KAONLT

- I continued restructure of this directory
  - There is a python package that handles a lot of what you're about to see behind the scenes

```
batch  config  HISTOGRAMS  OUTPUT  README.md  ROOTfiles
bin    DB        online_archive  README  REPORT OUTPUT  scripts
```

```
__init__.py  __init__.pyc  __pycache__  root2py.py  root2py.pyc
```

```
Branch-[offline_2020]
trottar ~/Analysis/hallc_replay_lt/UTIL_KAONLT/bin/python/root2py> █
```

```
lumi.cuts  pid.cuts  pid.cuts.tmp
```

```
Branch-[offline_2020]
trottar ~/Analysis/hallc_replay_lt/UTIL_KAONLT/DB/CUTS> █
```

# Python package: root2py

- My python package I created
- Can apply cuts, read in cuts, create nice density plots, convert csv to root file.
  - Planning to add equations to the package as well.
  - Eventually fitting too

```
# p_track_before Apply cuts
p_track_before = c.add_cut(P_dc_ntrack, "p_track_before")

# p_hadtrack_before
p_hadtrack_before = c.add_cut(P_dc_ntrack, "p_hadtrack_before")

# p_pitrack_before
p_pitrack_before = c.add_cut(P_dc_ntrack, "p_pitrack_before")

# p_Ktrack_before
p_Ktrack_before = c.add_cut(P_dc_ntrack, "p_Ktrack_before")
```

```
f = open('../../../../DB/CUTS/lumi.cuts')
# read in cuts file and make dictionary
c = r2p.pyPlot(None)
readDict = c.read_dict(f)

def make_cutDict(cut, f, inputDict=None):

    global c

    c = r2p.pyPlot(readDict) Call cuts file
    x = c.w_dict(cut)

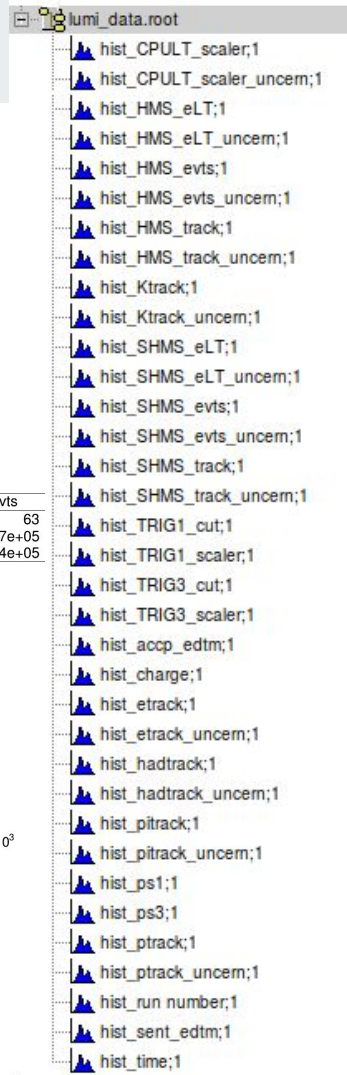
    if inputDict == None:
        inputDict = {}

    for key, val in readDict.items():
        if key == cut:
            inputDict.update({key : {}})

    for i, val in enumerate(x):
        tmp = x[i]
        inputDict[cut].update(eval(tmp))

    return inputDict

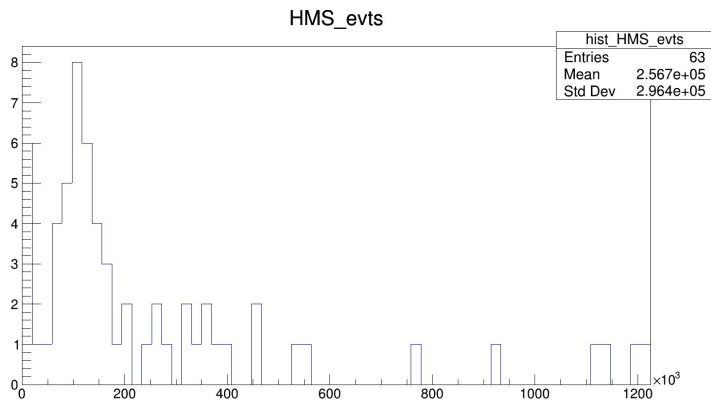
cutDict = make_cutDict("p_track_before", f)
cutDict = make_cutDict("p_hadtrack_before", f, cutDict)
cutDict = make_cutDict("p_Ktrack_before", f, cutDict)
cutDict = make_cutDict("p_pitrack_before", f, cutDict)
cutDict = make_cutDict("p_track_after", f, cutDict)
cutDict = make_cutDict("p_hadtrack_after", f, cutDict)
cutDict = make_cutDict("p_pitrack_after", f, cutDict)
cutDict = make_cutDict("p_Ktrack_after", f, cutDict)
cutDict = make_cutDict("p_pitrack_after", f, cutDict)
cutDict = make_cutDict("p_ecut_before", f, cutDict)
cutDict = make_cutDict("p_ecut_after", f, cutDict)
cutDict = make_cutDict("p_ecut_eff", f, cutDict)
cutDict = make_cutDict("h_track_before", f, cutDict)
cutDict = make_cutDict("h_etrack_before", f, cutDict)
cutDict = make_cutDict("h_track_after", f, cutDict)
cutDict = make_cutDict("h_etrack_after", f, cutDict)
cutDict = make_cutDict("h_etrack_after", f, cutDict)
cutDict = make_cutDict("h_ecut_after", f, cutDict)
cutDict = make_cutDict("h_ecut_eff", f, cutDict)
c = r2p.pyPlot(cutDict)
```



# Python package: root2py

- Pid and lumi directories have a csv2root.py file to convert csv to root histograms (root trees will eventually be used)

```
try:  
    pid_data = dict(pd.read_csv(inp_f))  
except IOError:  
    print("Error: %s does not appear to exist." % inp_f)  
print(pid_data.keys())  
r.py2root(pid_data,out_f)
```



# End goal

- Stephen and I have had a lot of conversations about where we want this to be
- The end goal is to combine his database files with my python package so that analysis is flexible, easy, and intuitive
- The cuts file will call Stephen's parameter files from a method in root2py
  - This will be per run with flags for run type (e.g. coin, singles)
- [https://github.com/trottar/hallc\\_replay\\_kaonlt/tree/offline\\_2020/UTIL\\_KAONLT](https://github.com/trottar/hallc_replay_kaonlt/tree/offline_2020/UTIL_KAONLT)

```
h_ecut_cer = {"H_gtr_dp" : (abs(H_gtr_dp) < 10.0)}, {"P_gtr_dp" : ((P_gtr_dp > -10.0) | (P_gtr_dp < 20.0))}, {"P_cal_etotnorm" : (P_cal_etotnorm < 0.6)}, {"H_gtr_beta" : ((abs(H_gtr_beta) - 1.00) < 0.1)}, {"CTime_eKCoinTime_ROC1" : (((CTime_eKCoinTime_ROC1 - 47.5) > -0.5) & ((CTime_eKCoinTime_ROC1 - 47.5) < 0.5))}, {"H_cal_etotnorm" : ((H_cal_etotnorm > 0.995) & (H_cal_etotnorm < 1.015))}, {"H_cer_npeSum" : (H_cer_npeSum > 3.0)}
#
h_ecut_no_cal = {"H_gtr_dp" : (abs(H_gtr_dp) < 10.0)}, {"P_gtr_dp" : ((P_gtr_dp > -10.0) | (P_gtr_dp < 20.0))}, {"P_cal_etotnorm" : (P_cal_etotnorm < 0.6)}, {"H_gtr_beta" : ((abs(H_gtr_beta) - 1.00) < 0.1)}, {"CTime_eKCoinTime_ROC1" : (((CTime_eKCoinTime_ROC1 - 47.5) > -0.5) & ((CTime_eKCoinTime_ROC1 - 47.5) < 0.5))}, {"H_cer_npeSum" : (H_cer_npeSum > 1.5)}
#
```