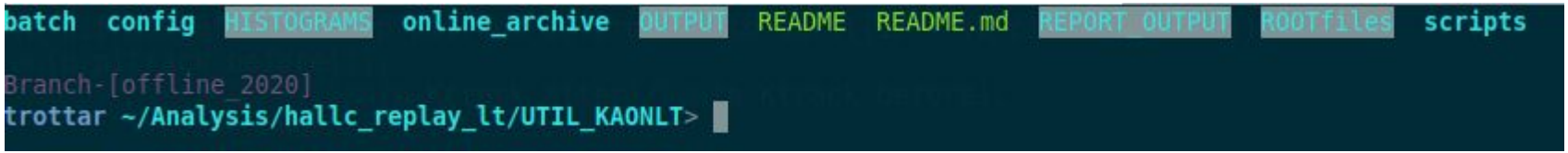
Kaon LT Status Update

April 15th, 2020

Richard Trotta

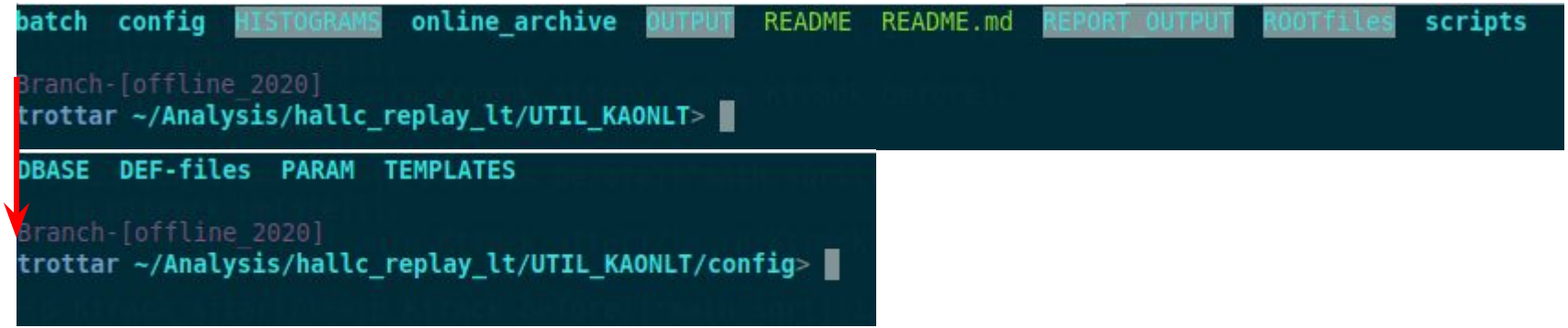
Restructured UTIL_KAONLT

- I restructured this directory to be explicitly used for all future analysis work
  - We can keep calibrations and other such studies in hallc_replay_lt as is



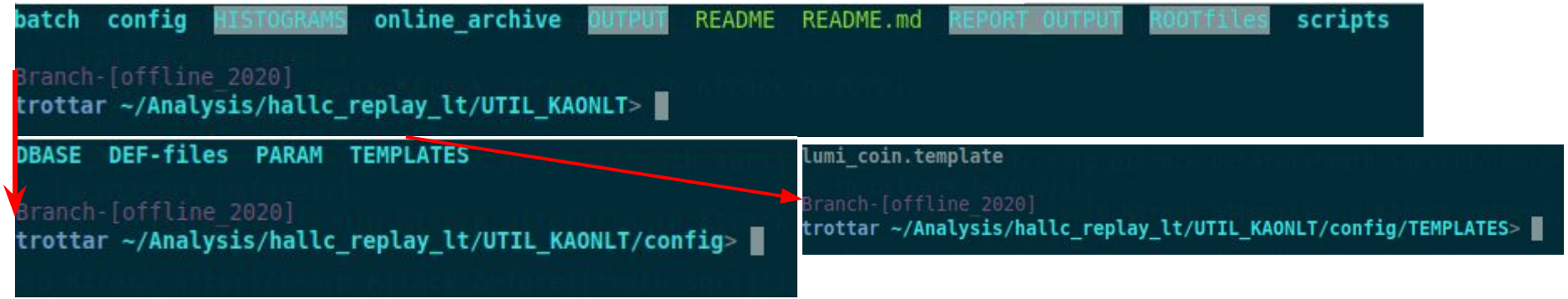
Restructured UTIL_KAONLT

- UTIL_KAONLT/config directory has all the configuration scripts (DBASE, DEF-files, PARAM, TEMPLATES)



Restructured UTIL_KAONLT

- UTIL_KAONLT/config directory has all the configuration scripts (DBASE, DEF-files, PARAM, TEMPLATES)
  - Primarily using DEF-files and TEMPLATES for now
  - DBASE and PARAM are more robust and better left in hallc_replay_lt



Restructured UTIL_KAONLT

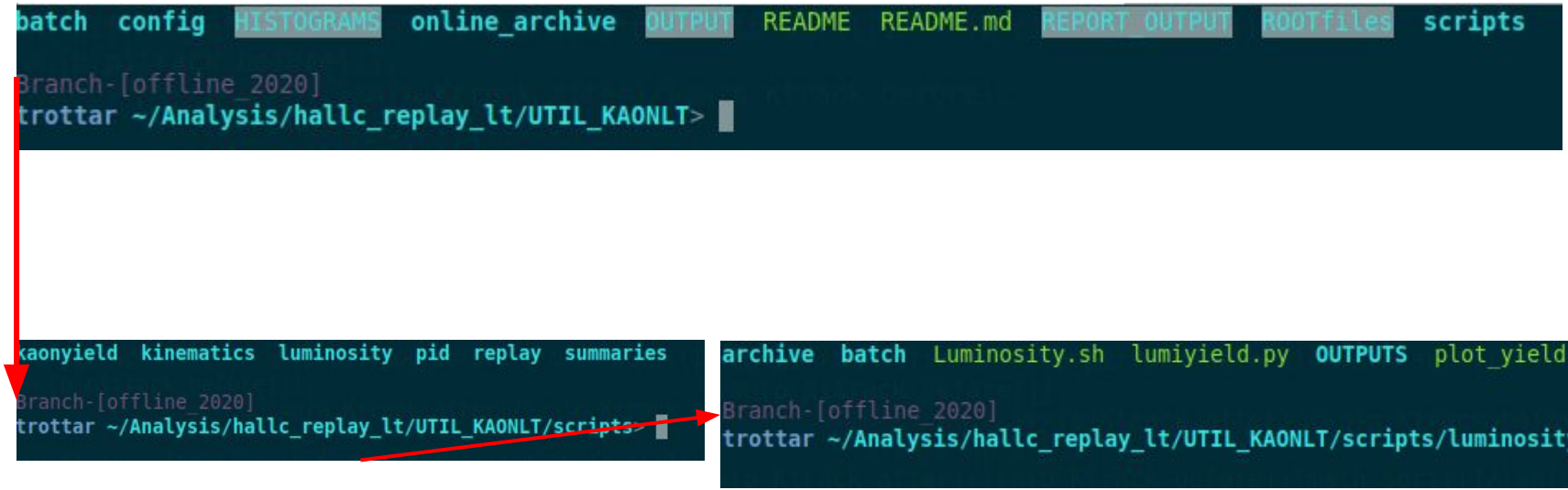
- UTIL_KAONLT/scripts contains all analysis type scripts

```
batch  config  HISTOGRAMS  online_archive  OUTPUT  README  README.md  REPORT_OUTPUT  ROOTfiles  scripts

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

```
kaonyield  kinematics  luminosity  pid  replay  summaries

Branch-[offline_2020]
trottar ~/Analysis/hallc_replay_lt/UTIL_KAONLT/scripts>
```

Restructured UTIL_KAONLT

- UTIL_KAONLT/scripts contains all analysis type scripts
  - I plan on structuring all these analysis script directories the same as the luminosity



New Luminosity Script

- Old C++/ROOT based script was giving lots of issues with newer ROOT files
- I rewrote the script in Python3 with more functionality (works in farm)
  - Python2 should work as well, but not tested
- There is the initial script lumiyield.py which applies cuts to various ROOT leaves and calculates things like tracking, HMS events, etc.
  - This will output a csv file which can be imported to an excel sheet or used in the second Python script
- The second Python script is plot_yield.py which reads in the csv file and does the final yield calculations
  - This will produce a yield plot and output a comprehensive csv file of all pertinent information
- Found a slight bug, which I have hopefully fixed by the time meetings starts

Yield Calculations (old slide)

- Current is calculated per event with a threshold current of 2.5 uA for BCM4B [See Sangwa's Talk]
- I have not incorporated non-scaler EDTM based calculations so no electronic livetime included
- Yield is calculated by

Let's look into how these four values are calculated

$$Y = \frac{N \times PS}{Q \times \varepsilon \times (cpuLT)} \pm \frac{\sqrt{N}}{N} Y$$

  - N is number of reconstructed events passing cuts, PS is the prescale value, and ε are tracking efficiencies

Number of Events (old slide)

- Two types of event selection
  - Using the event type leaf fEvtHdr.fEvtTyp where EvtType = 1 or 3 is a SHMS event and EvtType= 2 or 3 is a HMS event (previously used method)
  - Applying proper cuts to the TDC leaves to get the SHMS (3of4 in T.coin.pTRIG1_ROC2.tdcTime) and HMS (elreal in T.coin.pTRIG3_ROC1.tdcTime) event selection (more on this later)
  - In the end, these should result in the same event selection as long as fEvtHdr.fEvtTyp leaf is properly selecting events
- A number of cuts were applied as well
  - SHMS cuts: P_cal_etotnorm > 0.05, P_hgcer_npeSum > 1.5, P_aero_npeSum > 1.5
  - HMS cuts: H_cal_etotnorm > 0.6, H_cal_etotnorm < 2.0, H_cer_npeSum > 2.0

CPU Livetime (old slide)

- Originally this was calculated through purely scalers
  - cpuLT = L1Acc/[(ptrig1/ps1)+(ptrig3/ps3)]
- To improve this beyond the level one accepts the TDC leaves (described above) were used
  - The same cuts as the event selection were applied
  - cpuLT = (TDC_trig1cut+TDC_trig3cut)/[(ptrig1-EDTM/ps1)+(ptrig3-EDTM/ps3)]
- The latest improvement was to separate the HMS cpuLT and SHMS cpuLT and calculate them separately
  - cpuLT_HMS = TDC_trig3cut/[(ptrig3-EDTM)/ps3]
  - cpuLT_SHMS = TDC_trig1cut/[(ptrig1-EDTM)/ps1]

Tracking Efficiencies (old slide)

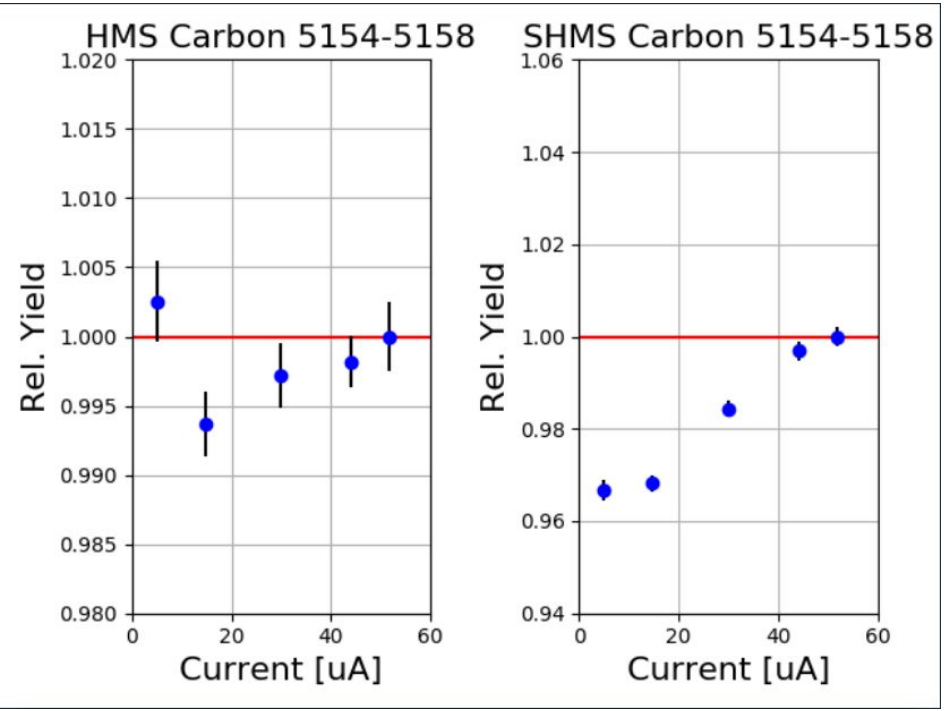
- The runs that I looked at had electrons in the HMS and pions in the SHMS
  - P_{HMS}=-3.266, Θ_{HMS}=12.50, P_{SHMS}=+6.842,Θ_{SHMS}=6.55
  - P_{HMS}=-4.204, Θ_{HMS}=14.51 P_{SHMS}=+6.053,Θ_{SHMS}=6.55
- HMS tracking was found by applying cuts to H.dc.ntrack
- SHMS tracking was found by applying cuts to P.dc.ntrack
- The HMS used the electron tracking efficiency while the SHMS used the pion tracking efficiency (note that originally the SHMS used the hadron tracking efficiency but there was little change going to pion tracking)

Tracking Efficiencies (con't) (old slide)

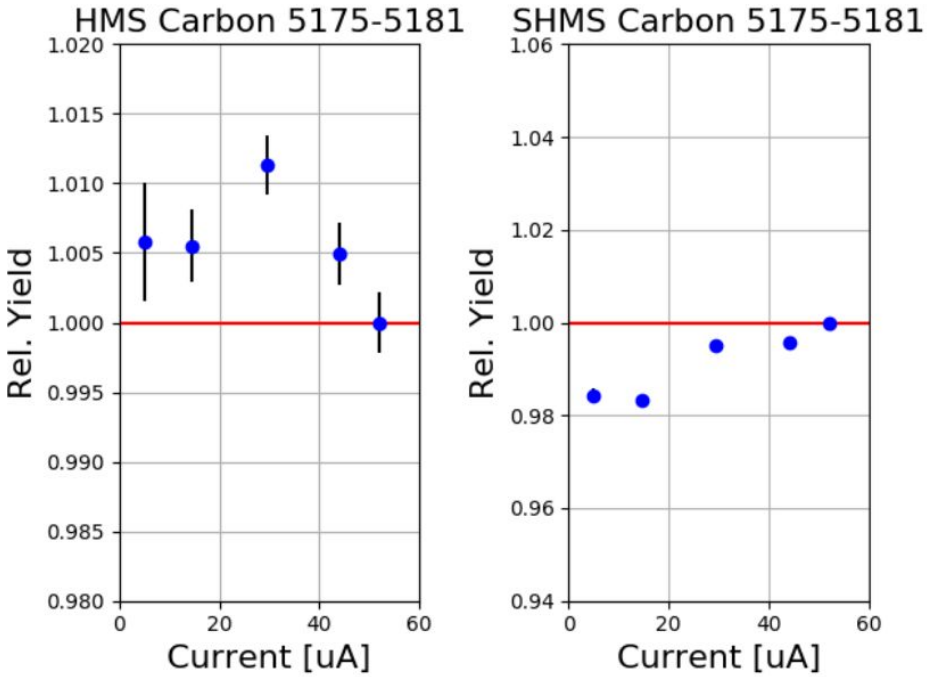
- Electron tracking efficiency cuts -> H_hod_goodscinhit == 1 & H_hod_betanotrack > 0.8 & H_hod_betanotrack < 1.3 & (H_dc_1x1_nhit + H_dc_1u2_nhit + H_dc_1u1_nhit + H_dc_1v1_nhit + H_dc_1x2_nhit + H_dc_1v2_nhit) < 20 & (H_dc_2x1_nhit + H_dc_2u2_nhit + H_dc_2u1_nhit + H_dc_2v1_nhit + H_dc_2x2_nhit + H_dc_2v2_nhit) < 20 & H_cer_npeSum > 0.5 & H_cal_etotnorm > 0.6 & H_cal_etotnorm < 2.0
- Pion tracking efficiency cuts ->P_hod_goodscinhit == 1 & P_hod_betanotrack > 0.5 & P_hod_betanotrack < 1.4 & (P_dc_1x1_nhit + P_dc_1u2_nhit + P_dc_1u1_nhit + P_dc_1v1_nhit + P_dc_1x2_nhit + P_dc_1v2_nhit) < 20 &(P_dc_2x1_nhit + P_dc_2u2_nhit + P_dc_2u1_nhit + P_dc_2v1_nhit + P_dc_2x2_nhit + P_dc_2v2_nhit) < 20 & P_cal_etotnorm > 0.05 & P_cal_etotnorm <= 0.6 & P_hgcer_npeSum > 10 & P_aero_npeSum > 3
- Hadron tracking efficiency cuts ->P_hod_goodscinhit == 1 & P_hod_betanotrack > 0.5 & P_hod_betanotrack < 1.4 & (P_dc_1x1_nhit + P_dc_1u2_nhit + P_dc_1u1_nhit + P_dc_1v1_nhit + P_dc_1x2_nhit + P_dc_1v2_nhit) < 20 &(P_dc_2x1_nhit + P_dc_2u2_nhit + P_dc_2u1_nhit + P_dc_2v1_nhit + P_dc_2x2_nhit + P_dc_2v2_nhit) < 20 & P_cal_etotnorm > 0.05 & P_cal_etotnorm <= 0.6

Carbon plots (old slide)

P_{HMS}=-3.266, Θ_{HMS}=12.50 P_{SHMS}=+6.842,Θ_{SHMS}=6.55



P_{HMS}=-4.204, Θ_{HMS}=14.51 P_{SHMS}=+6.053,Θ_{SHMS}=6.55



New PID script

- Like the lumi script, old C++/ROOT based script was giving lots of issues with newer ROOT files
  - Although issues were not as ubiquitous
- In the process of converting this script over, should take two or three days to complete and test
- Eventually I would like to convert all scripts that use TProof as this has been the source of all the issues for myself
  - Of course, all c++ versions will be kept in UTIL_KAONLT/script/<analysis>/archive for others use if they prefer