AliExternalInfo

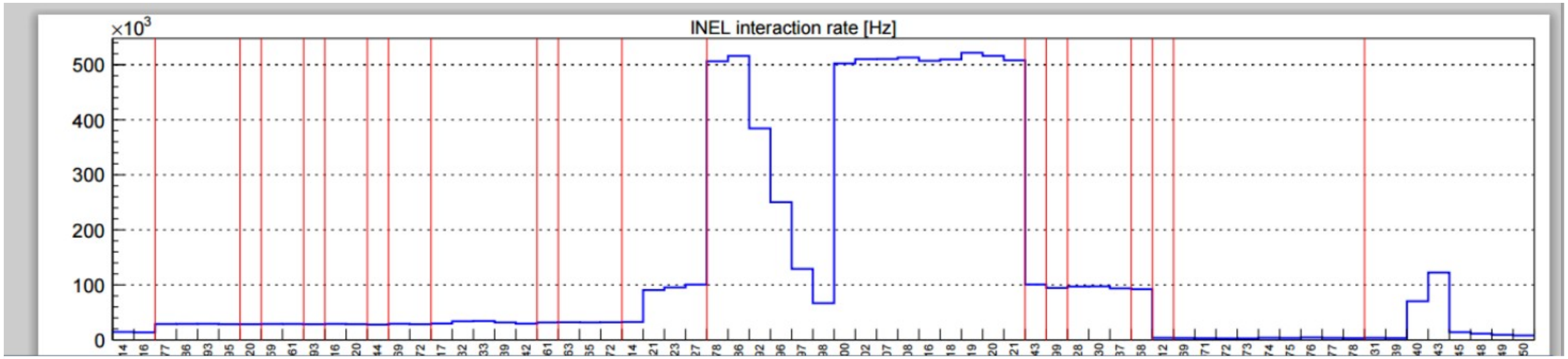
Accessing resources (DAQ, RCT, detector QA, etc...)

Carsten Klein
Goethe Universität Frankfurt

With Jens Wiechula & Marian Ivanov

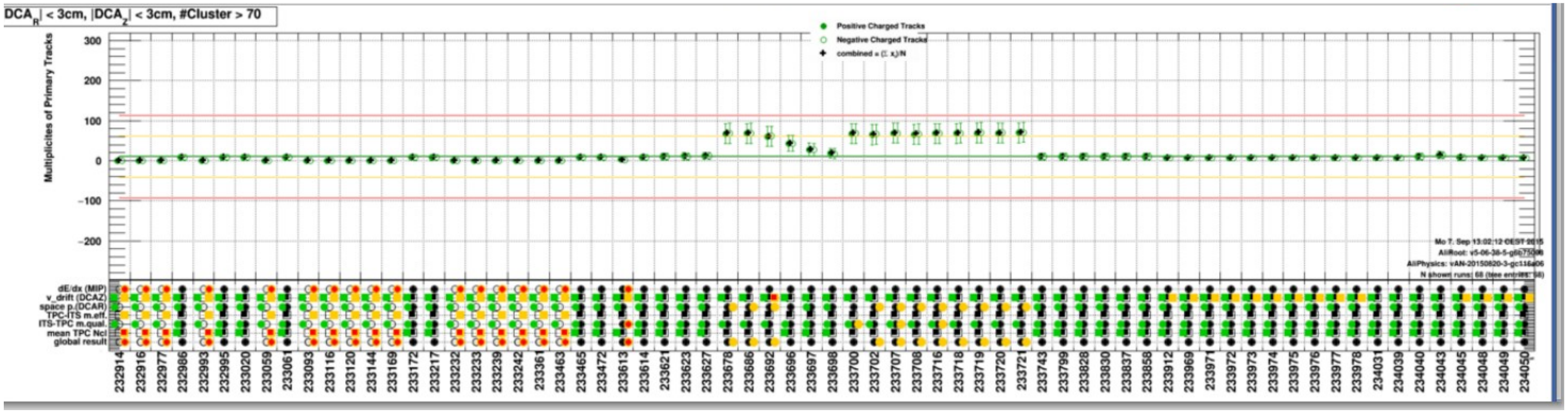
First Idea

Interaction Rate



http://aliqaevs.web.cern.ch/aliqaevs/data/2015/LHC15h/pass1/global_properties.pdf
http://aliqatpc.web.cern.ch/aliqatpc/data/2015/LHC15h/pass1/meanMult_vs_run.png

Multiplicity



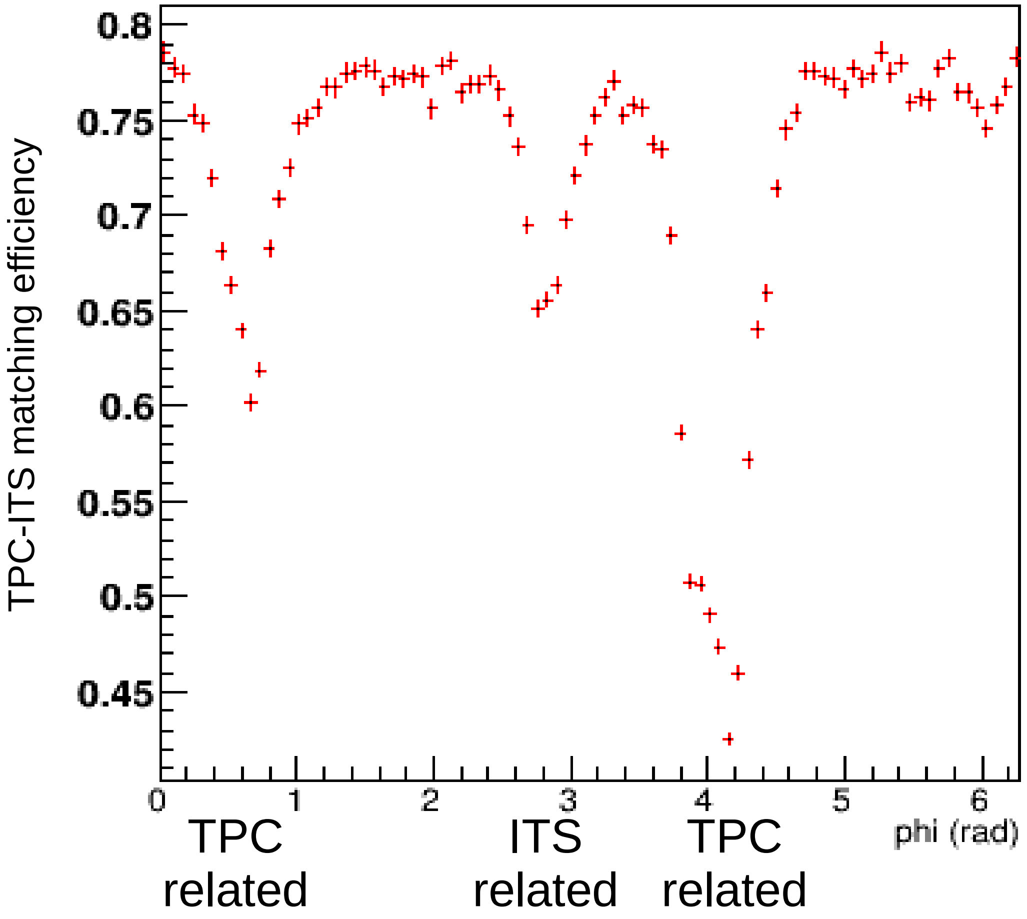
- High interaction rate leads to pile-up events in the TPC
- Start of a new fill (with higher IR) easily visible in the TPC QA plot “Multiplicity”
- Higher interaction rate leads to a worse performance of the TPC
- **Correct multiplicities as a function of the interaction rate**
  - Source of Information: **EVS QA**
- Additional remark: Space charge distortions increase with increasing interaction rate

Second Idea

- TPC QA often sensitive to different detectors, like ITS
- Where do those holes come from?
- Getting information from ITS will directly tell us, if ITS has parts of the detector off
  - Source of information: **RCT / ITS QA**

==> We need to have a tool/class which gathers all the information which is available in ALICE and create trees and chains (by combining several similar trees) for an easy usage
==> AliExternalInfo

LHC15n - Run 244416 – IR 100 kHz

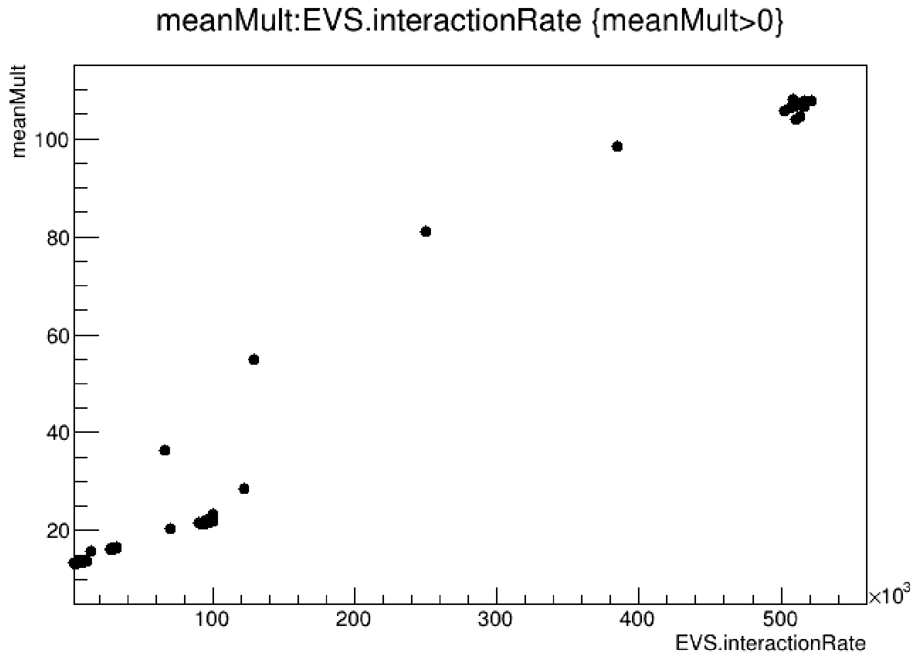


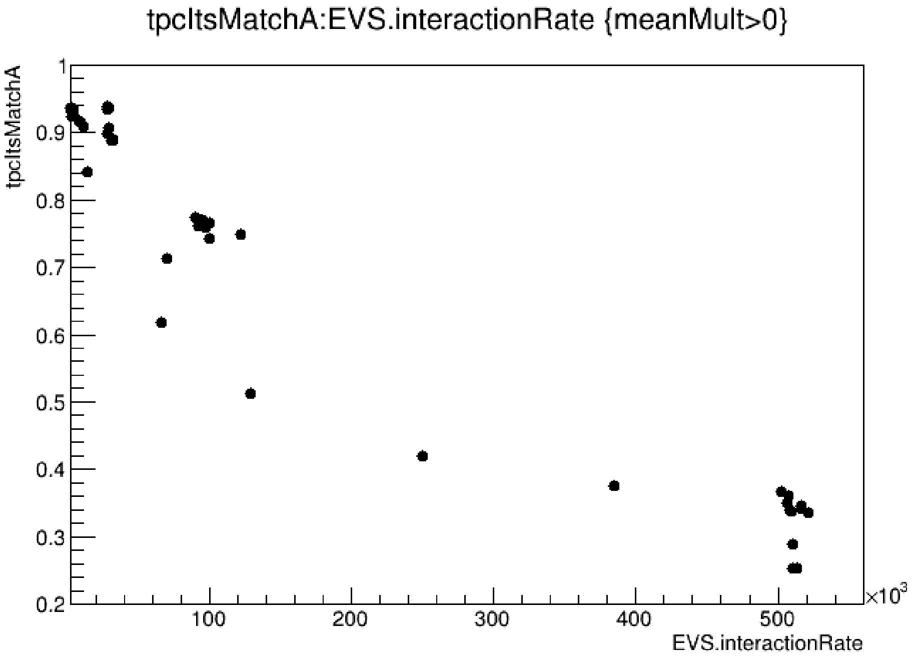
Status

- Download several resources (could be done via cronjob)
  (logbook, RCT, MC, Detector QA, MonALISA, etc…)
- Generate a tree from those resources
- Create a chain from those resources
- Set a global time limit for downloaded resources
- Set storage location on your disk

AliExternalInfo in Action

```
AliExternalInfo inf;
TTree* trendingEVS = inf.GetTreeDataQA("EVS","LHC15h","pass1");
TTree* trendingTPC = inf.GetTreeDataQA("TPC","LHC15h","pass1");
trendingEVS->BuildIndex("run");
trendingTPC->BuildIndex("run");
trendingTPC->AddFriend(trendingEVS,"EVS");
trendingTPC->SetMarkerStyle(20);
trendingTPC->SetMarkerSize(1);
trendingTPC->Draw("meanMult:EVS.interactionRate","meanMult>0");
trendingTPC->Draw("tpcitsMatchA:EVS.interactionRate","meanMult>0");
```





Syntax I

- Constructor:
```
AliExternalInfo infoObj(TString Globalpath = “.”); // Where to store the trees
```

- Download of the resource in a directory structure:
```
Globalpath/data/<year>/<period>/<pass>/ or
Globalpath/sim/ or respectively:


infoObj.CacheMC(); // Downloading of the MC database
infoObj.CacheRCT(TString period, TString pass); // Download a specific RCT page
infoObj.CacheLogbook(TString period); // Download a specific logbook entry
infoObj.CacheTriggerClasses(TString year);
infoObj.CacheDataQA(TString detector, TString period, TString pass); // det QA
infoObj.CacheProdCycle(); // from: https://alimonitor.cern.ch/prod/?t=3
infoObj.CacheProdCycle(TString id); // information for a specific production
infoObj.CacheProdPasses(); // from: https://alimonitor.cern.ch/prod/?t=1
```

- Set global time limit for files:
```
infoObj.SetGlobalTimeLimit(long int t); // in seconds
```

Syntax II

- Create a tree out of the downloaded data, if the data is not yet downloaded it will also download it:

```
TTree* treeRCT = infoObj.GetTreeRCT(TString period, TString pass);
etc…
```

- Create a chain from those trees:

```
TTree* tree10dRCT = b.GetTreeRCT(“LHC10d”, “pass2”);
TTree* tree10eRCT = b.GetTreeRCT(“LHC10e”, “pass2”);
TTree* tree10fRCT = b.GetTreeRCT(“LHC10f”, “pass4”);
infoObj.GetChainRCT(“LHC10*”, “pass*”);
```

- Note the asterisk in the chain function which allows to chain several periods/passes

Outlook and Plans

- Commit to AliRoot (next couple of days)
- Code cleanup
- Configuration file:
  - Set the location of the external resources
  - Set the timer for every resource individually
- Chains: Comma separated list for periods and passes, like “LHC10c, LHC10d” or search by regular expressions
- Add many more resources (input from you would be nice)
- Therefore it would be nice to agree on common formats throughout all resources:
  - Place where the trending.root is stored
  - Name of the tree inside
  - Granularity (Per period or per pass?)
  - Minimal set of variables inside (e.g. run number with proper branch names)
  - Probably a discussion with experts in the e.g. weekly calibration meeting