DT Efficiency monitoring versus integrated luminosity

Claudio Caputo [1],

[1] Università degli Studi di Bari, INFN Bari

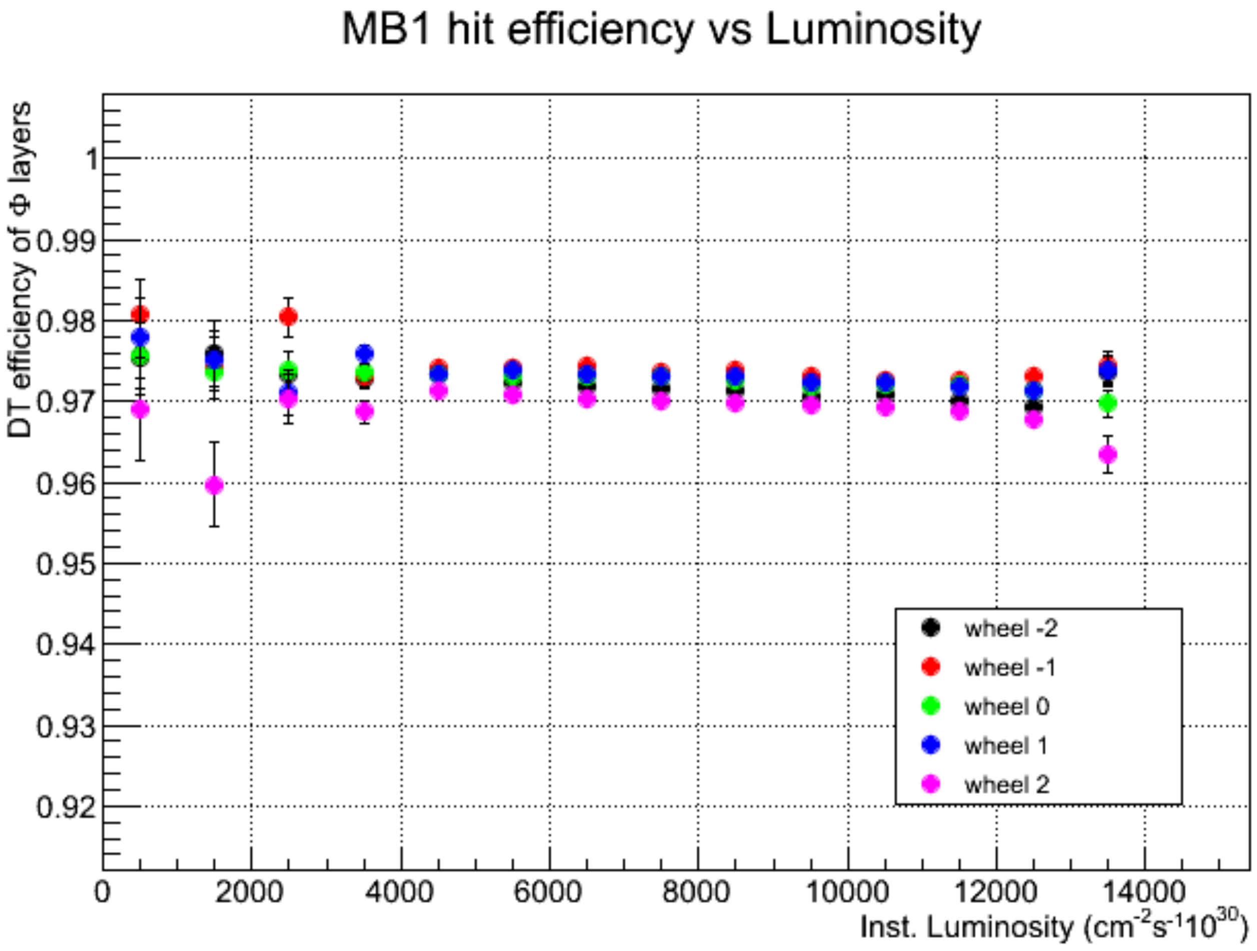
New Muon DOC3s shift

- Original motivation was the restructuring of central offline shifts. In particular, Offline DQM shifts are suppressed this year.
- Muon project requested and obtained two new shift positions called DOC3 to start in 2017:
    - DOC3#1 - Trend monitoring
    - DOC3#2 - Data Certification

More info in L. Guiducci slides (link)

Trend monitoring

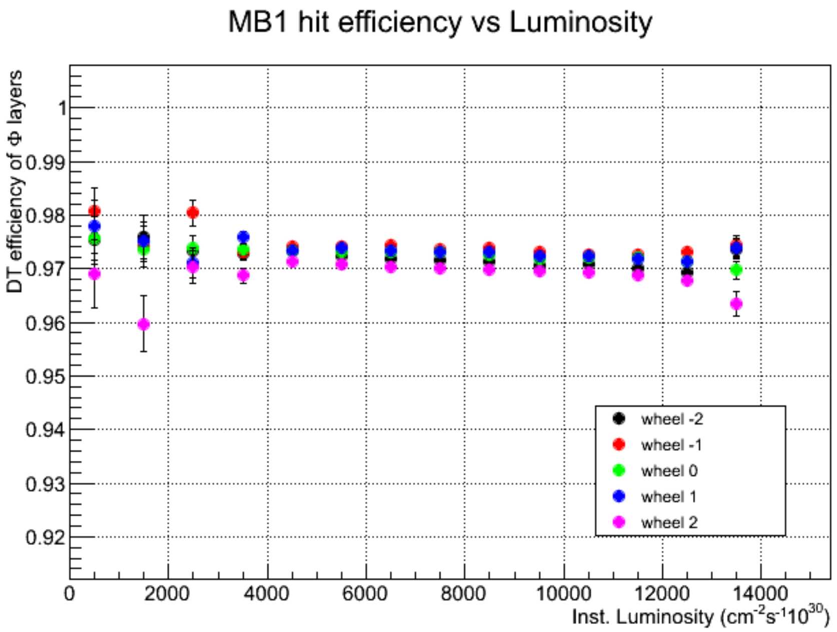
- Several tools for trend monitoring exist or are being developed
- Some development still needed, especially for long-term stability monitoring (e.g. integrated lumi)



DT efficiency plots **vs Integrated Lumi** for trend monitoring will be shown

Starting point - DPG DT trend plots

- Code for DT trend plots already available (F. Cavallo)
  - provides the logic for the efficiencies computation
  - Vs inst. Luminosity, PU, Run Number
- Use this code as a starting point for Eff Vs Int. Lumi
- 2016 Zmumu Dataset available:



"Run2016B": “/eos/cms/store/group/dpg_dt/comm_dt/dtRootple2016/Run2016BZMu23Sep2016-v1.root",
"Run2016C": "/eos/cms/store/group/dpg_dt/comm_dt/dtRootple2016/Run2016CZMu23Sep2016-v1.root",
"Run2016D": "/eos/cms/store/group/dpg_dt/comm_dt/dtRootple2016/Run2016DZMu23Sep2016-v1.root",
"Run2016E": "/eos/cms/store/group/dpg_dt/comm_dt/dtRootple2016/Run2016EZMu23Sep2016-v1.root",
"Run2016F": "/eos/cms/store/group/dpg_dt/comm_dt/dtRootple2016/Run2016FZMu23Sep2016-v1.root",
"Run2016G": "/eos/cms/store/group/dpg_dt/comm_dt/dtRootple2016/Run2016GZMu23Sep2016-v1.root",
"Run2016H": "/eos/cms/store/group/dpg_dt/comm_dt/dtRootple2016/Run2016HZMuPromptReco-v2.root"

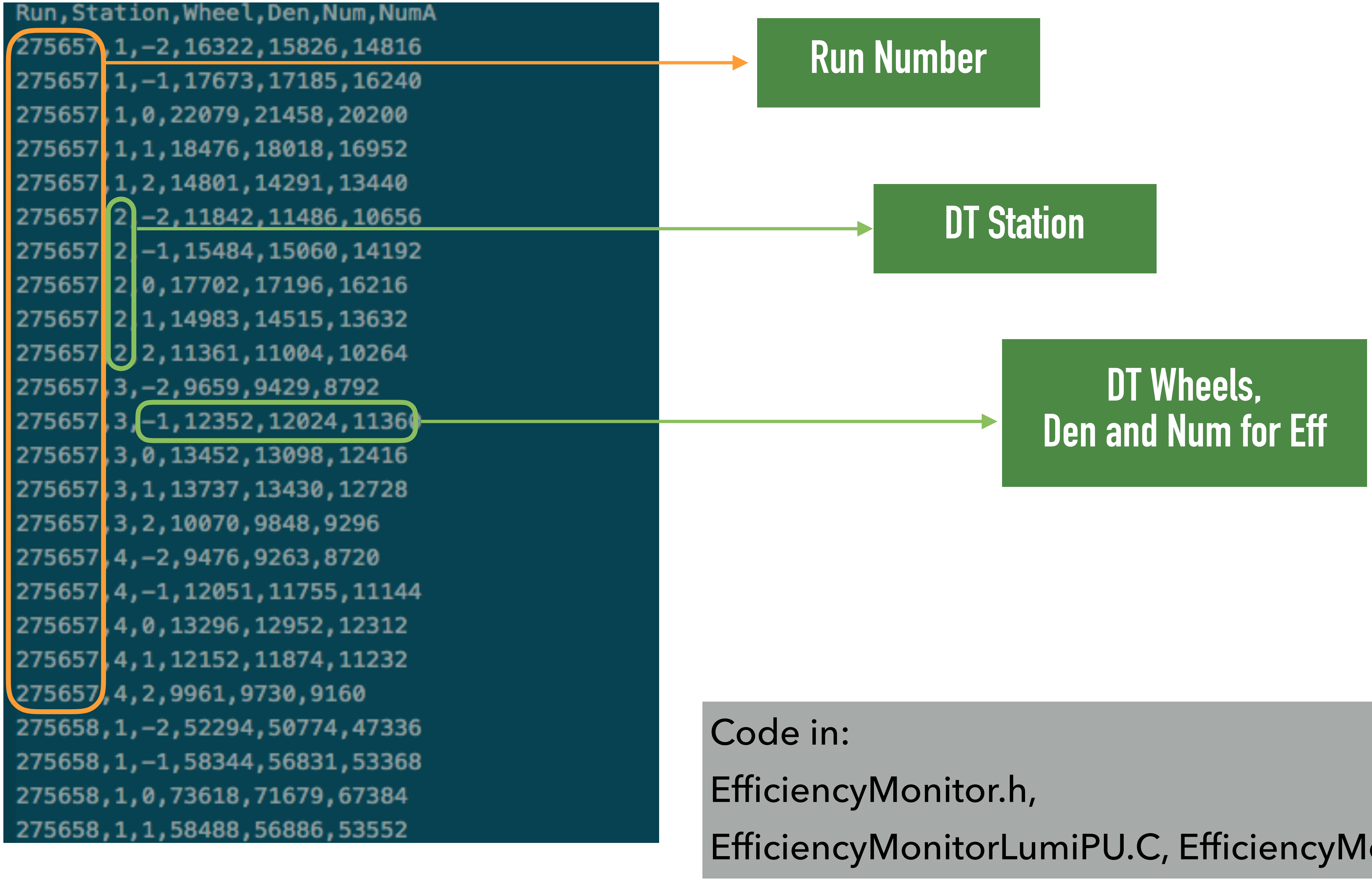
- RunC used for developing and testing the Eff Vs Int. Lumi plots

Strategy for Eff Vs int. Lumi plots

1. Store all quantities needed for Eff computation together per Run Number, DT station, DT wheel in a .csv file
2. Create a JSON file with the Run Numbers and lumi section contained in the sample used
3. Use brilcalc for int. Lumi info, saving them in a new .csv
4. Combine the Eff file with the int. Lumi file in order to have the int. Lumi value related to each Run Number
5. The final file contains all the informations needed for the Eff Vs int. Lumi plots and it could be:
    - open as a TTree
    - store in a database
    - read with different plotting tools

Strategy for Eff Vs int. Lumi plots

1. Store all quantities needed for Eff computation together per Run Number, DT station, DT wheel in a .csv file



```
Run,Station,Wheel,Den,Num,NumA
275657,1,-2,16322,15826,14816
275657,1,-1,17673,17185,16240
275657,1,0,22079,21458,20200
275657,1,1,18476,18018,16952
275657,1,2,14801,14291,13440
275657,2,-2,11842,11486,10656
275657,2,-1,15484,15060,14192
275657,2,0,17702,17196,16216
275657,2,1,14983,14515,13632
275657,2,2,11361,11004,10264
275657,3,-2,9659,9429,8792
275657,3,-1,12352,12024,11360
275657,3,0,13452,13098,12416
275657,3,1,13737,13430,12728
275657,3,2,10070,9848,9296
275657,4,-2,9476,9263,8720
275657,4,-1,12051,11755,11144
275657,4,0,13296,12952,12312
275657,4,1,12152,11874,11232
275657,4,2,9961,9730,9160
275658,1,-2,52294,50774,47336
275658,1,-1,58344,56831,53368
275658,1,0,73618,71679,67384
275658,1,1,58488,56886,53552
```

Run Number

DT Station

DT Wheels,
Den and Num for Eff

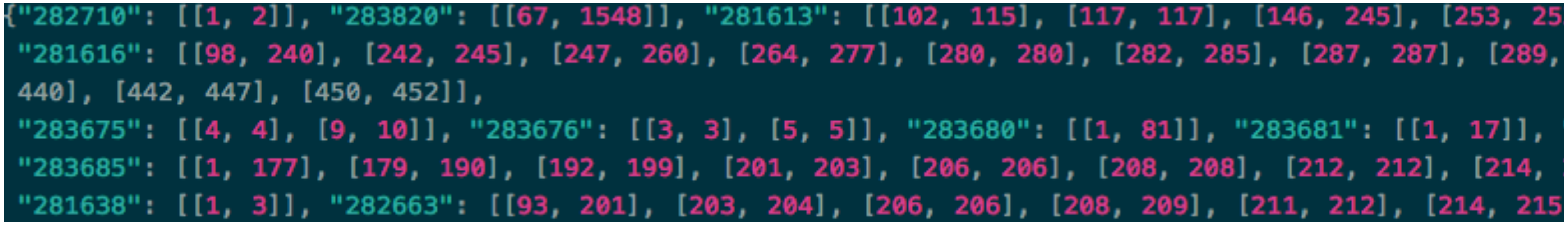
Code in:

EfficiencyMonitor.h,

EfficiencyMonitorLumiPU.C, EfficiencyMonitorRun.C

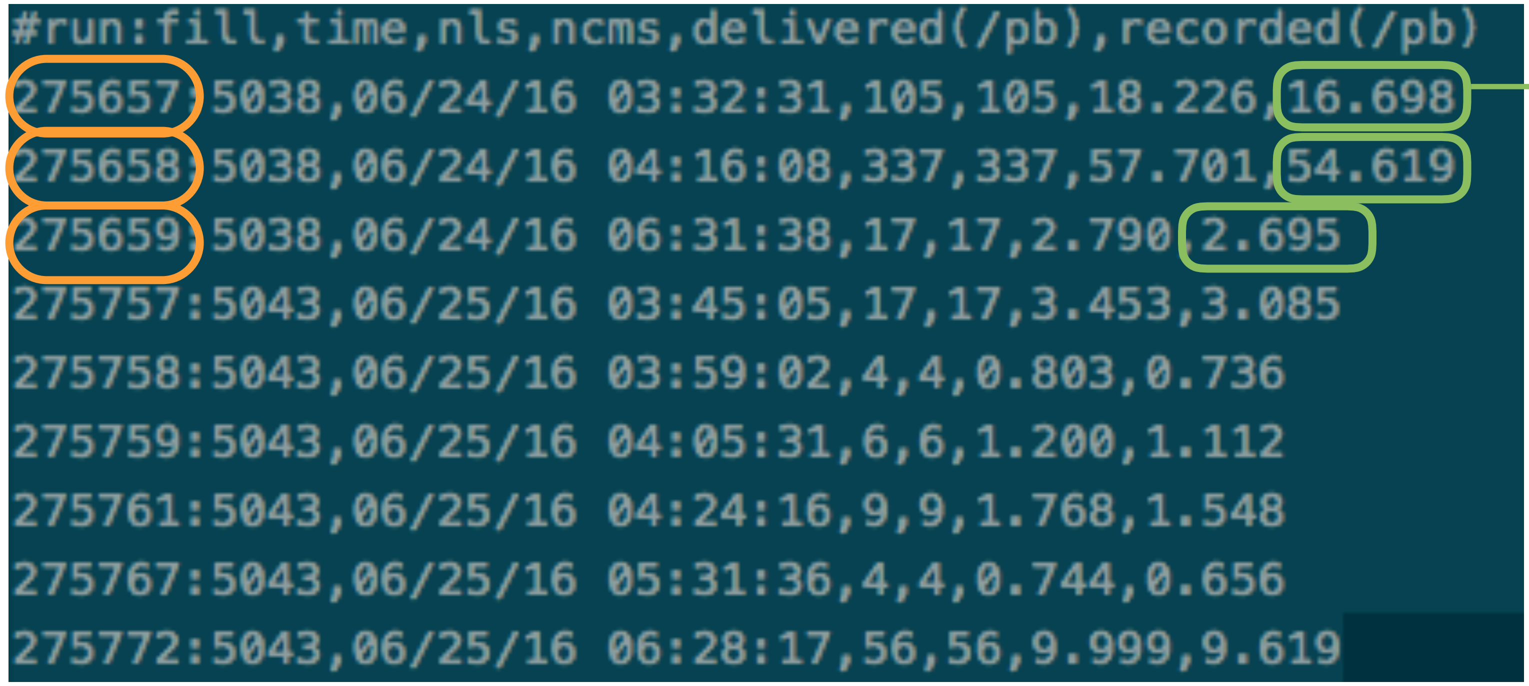
Strategy for Eff Vs int. Lumi plots

2. Create a JSON file with the Run Numbers and lumi section contained in the sample used



```
{"282710": [[1, 2]], "283820": [[67, 1548]], "281613": [[102, 115], [117, 117], [146, 245], [253, 25
 "281616": [[98, 240], [242, 245], [247, 260], [264, 277], [280, 280], [282, 285], [287, 287], [289,
 440], [442, 447], [450, 452]],
 "283675": [[4, 4], [9, 10]], "283676": [[3, 3], [5, 5]], "283680": [[1, 81]], "283681": [[1, 17]],
 "283685": [[1, 177], [179, 190], [192, 199], [201, 203], [206, 206], [208, 208], [212, 212], [214,
 "281638": [[1, 3]], "282663": [[93, 201], [203, 204], [206, 206], [208, 209], [211, 212], [214, 215
```

3. Use brilcalc for int. Lumi info, saving them in a new .csv



```
#run:fill,time,nls,ncms,delivered(/pb),recorded(/pb)
275657:5038,06/24/16 03:32:31,105,105,18.226,16.698
275658:5038,06/24/16 04:16:08,337,337,57.701,54.619
275659:5038,06/24/16 06:31:38,17,17,2.790,2.695
275757:5043,06/25/16 03:45:05,17,17,3.453,3.085
275758:5043,06/25/16 03:59:02,4,4,0.803,0.736
275759:5043,06/25/16 04:05:31,6,6,1.200,1.112
275761:5043,06/25/16 04:24:16,9,9,1.768,1.548
275767:5043,06/25/16 05:31:36,4,4,0.744,0.656
275772:5043,06/25/16 06:28:17,56,56,9.999,9.619
```

Int. Lumi recorded

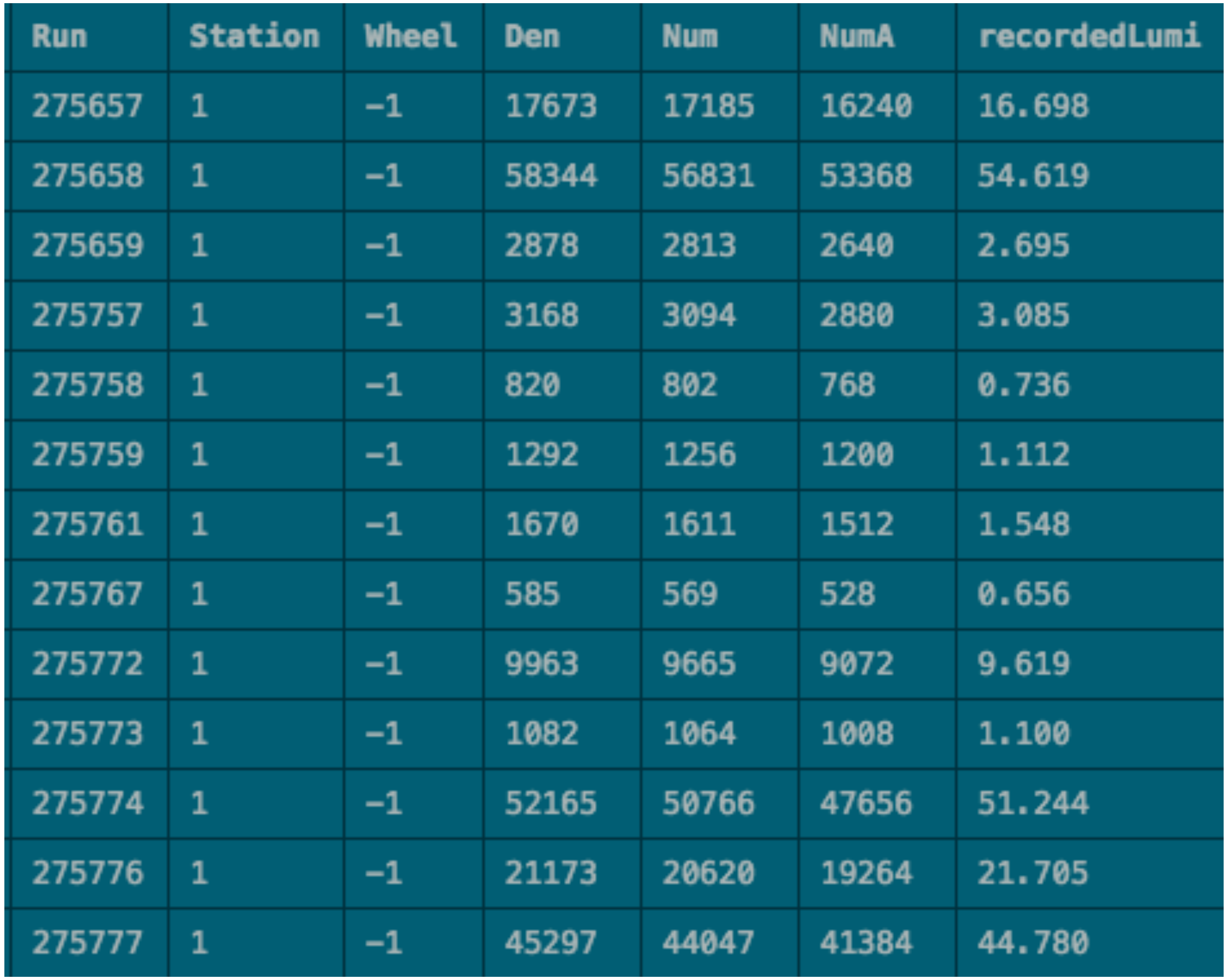
- standard **brilcalc** output
- the script for producing this file is tuned on the DT Tree but is easily editable for different samples

Code in:

createJSONs.py, lumi_utils-py

Strategy for Eff Vs int. Lumi plots

4. Combine the Eff file with the int. Lumi file in order to have the int. Lumi value related to each Run Number
    - using python modules (pandas, numpy)



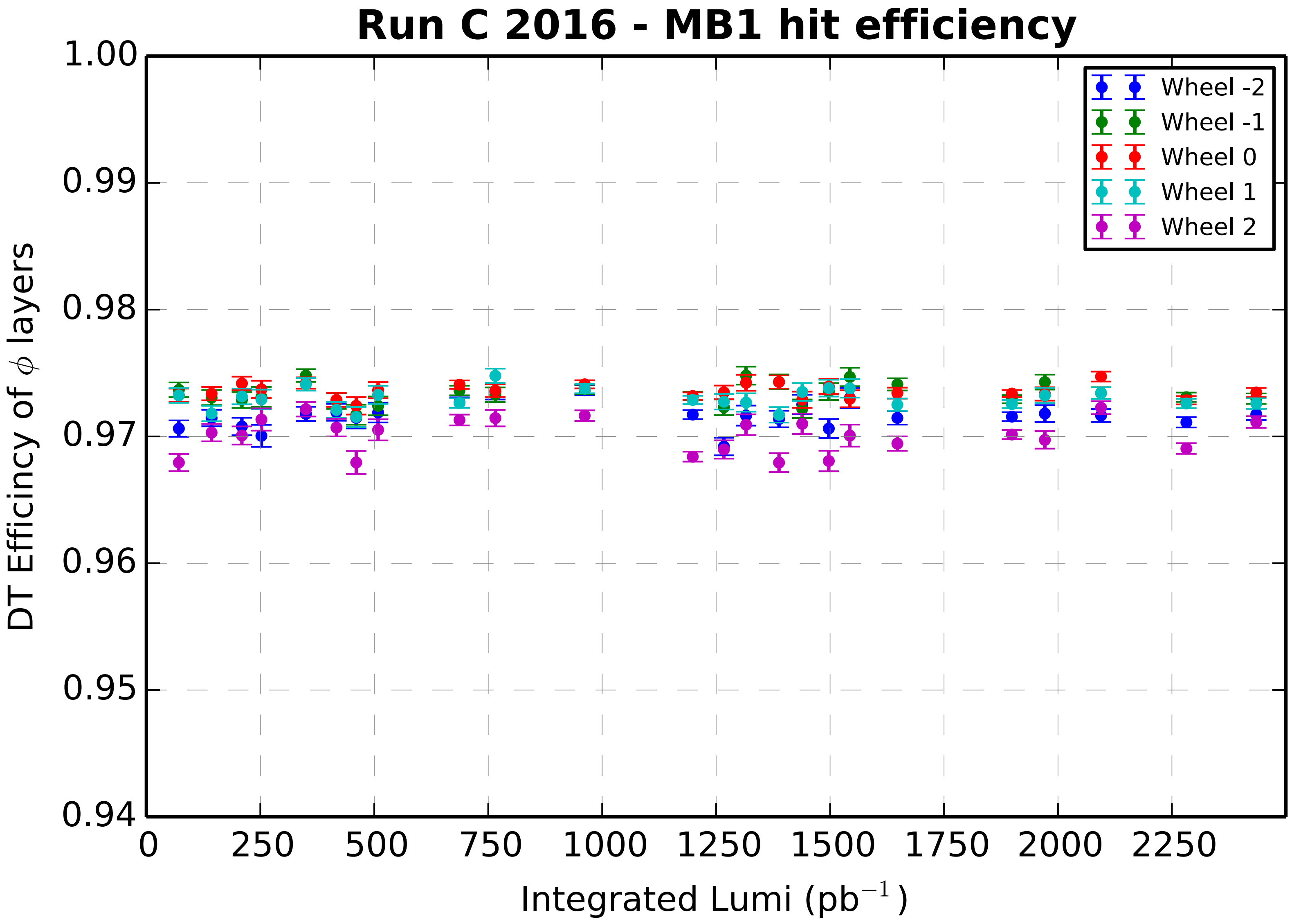
Run	Station	Wheel	Den	Num	NumA	recordedLumi
275657	1	-1	17673	17185	16240	16.698
275658	1	-1	58344	56831	53368	54.619
275659	1	-1	2878	2813	2640	2.695
275757	1	-1	3168	3094	2880	3.085
275758	1	-1	820	802	768	0.736
275759	1	-1	1292	1256	1200	1.112
275761	1	-1	1670	1611	1512	1.548
275767	1	-1	585	569	528	0.656
275772	1	-1	9963	9665	9072	9.619
275773	1	-1	1082	1064	1008	1.100
275774	1	-1	52165	50766	47656	51.244
275776	1	-1	21173	20620	19264	21.705
275777	1	-1	45297	44047	41384	44.780

- table for MB1, wheel -1
- the table could be stored in different formats
- the table could be extended with more columns (variables)

Code in:
effVsLumi.py

Eff Vs int. Lumi plot - Run C - DT Station 1

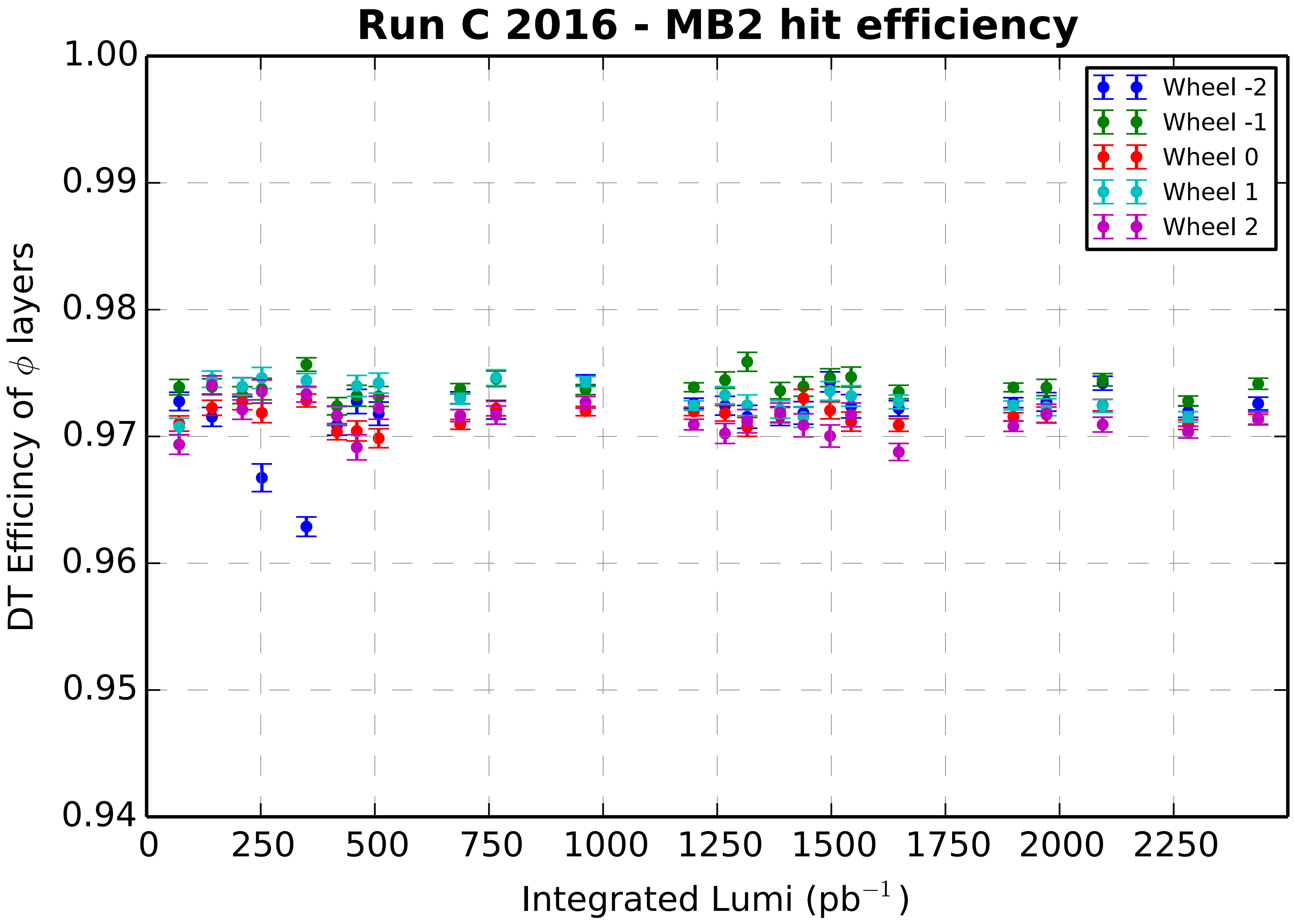
5. The final file contains all the informations needed for the Eff Vs int. Lumi plots.
   - Open with python modules (pandas, matplotlib)
   - “re-bin” in order to have enough statistics for Eff
     - re-bin could be tuned



Code in:
effVsLumi.py

Eff Vs int. Lumi plot – Run C – DT Station 2

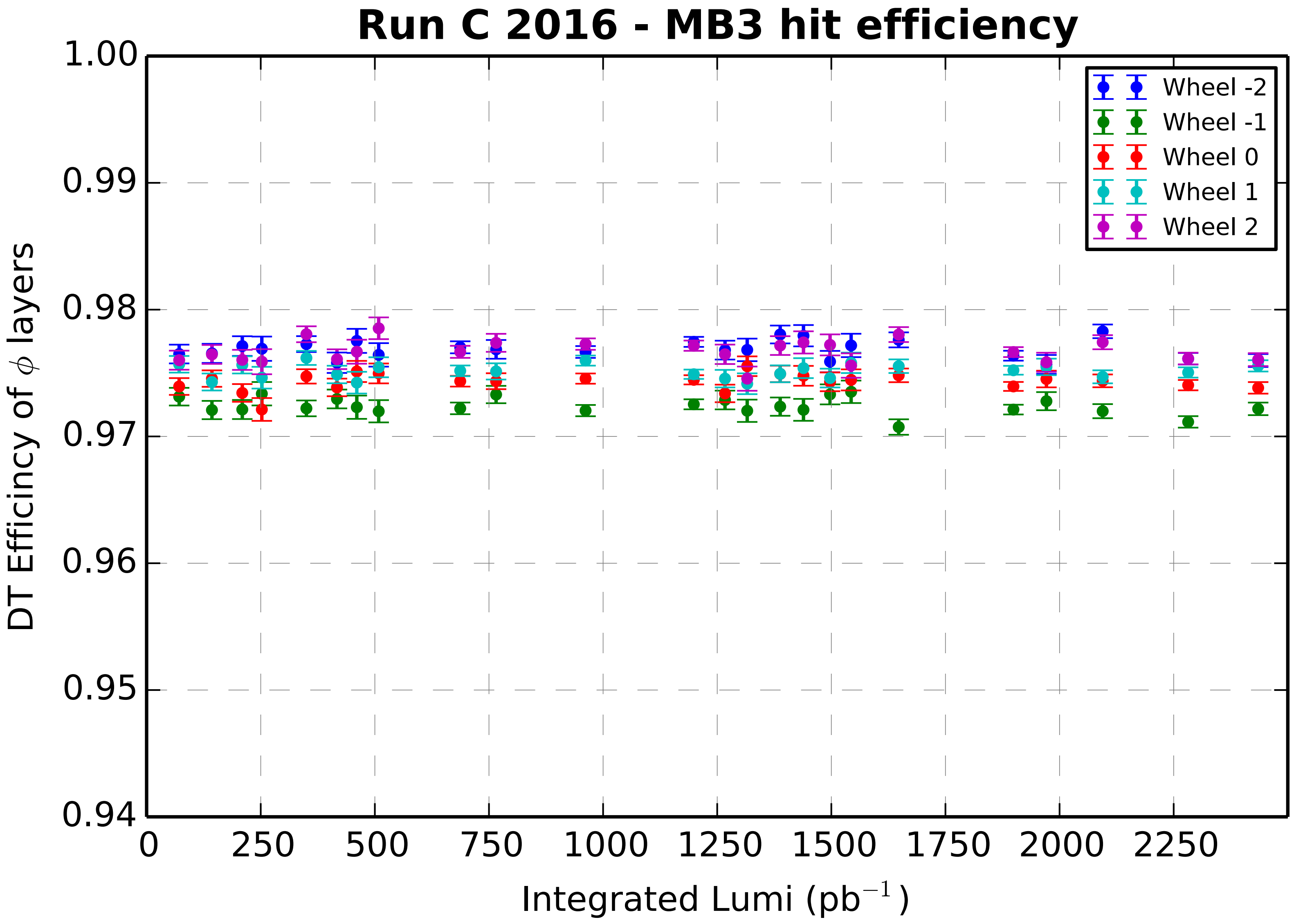
5. The final file contains all the informations needed for the Eff Vs int. Lumi plots.
   - Open with python modules (pandas, matplotlib)
   - “re-bin” in order to have enough statistics for Eff
     - re-bin could be tuned



Code in: effVsLumi.py

Eff Vs int. Lumi plot – Run C – DT Station 3

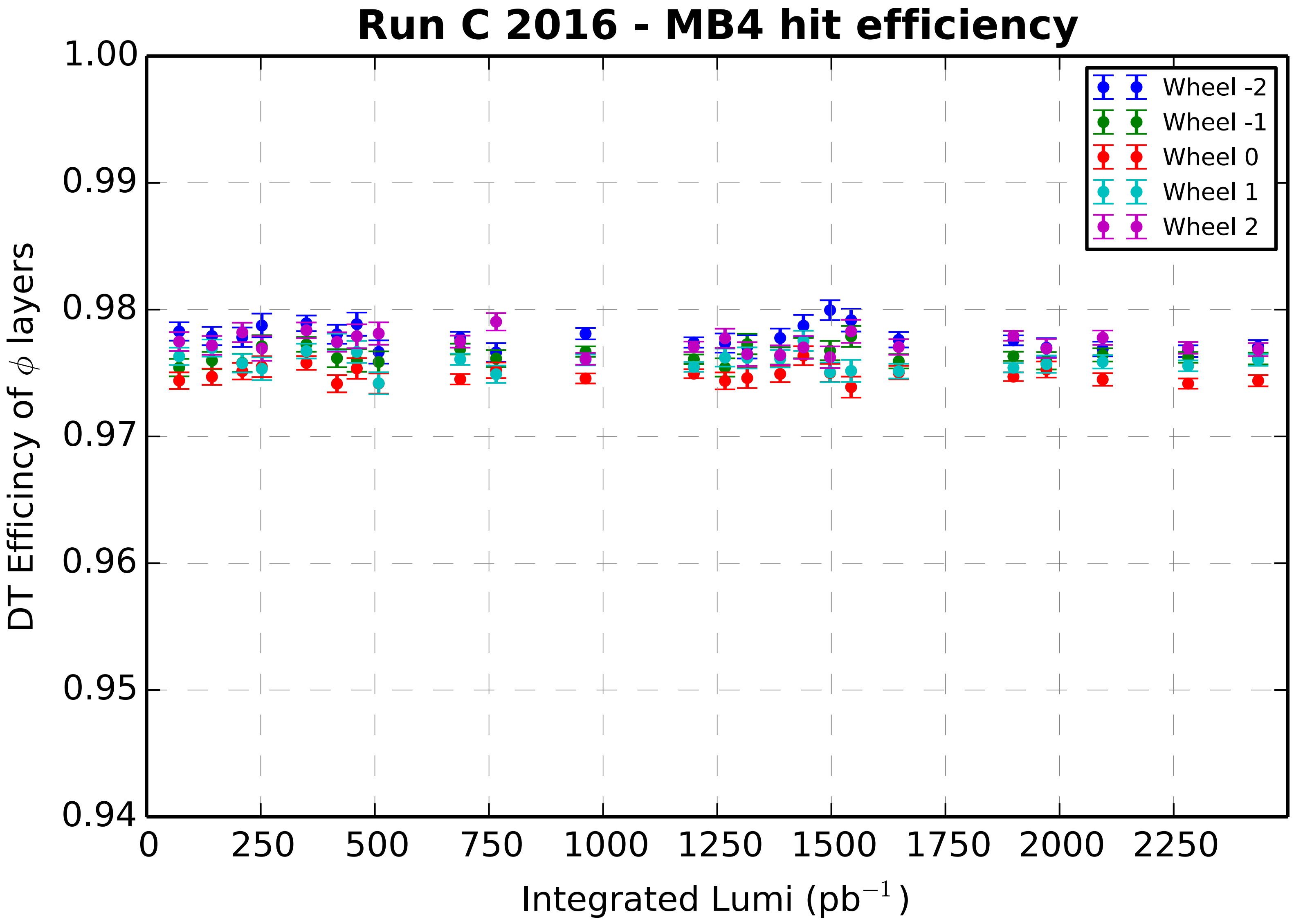
5. The final file contains all the informations needed for the Eff Vs int. Lumi plots.
   - Open with python modules (pandas, matplotlib)
   - "re-bin" in order to have enough statistics for Eff
     - re-bin could be tuned



Code in:
effVsLumi.py

Eff Vs int. Lumi plot - Run C - DT Station 4

5. The final file contains all the informations needed for the Eff Vs int. Lumi plots.
   - Open with python modules (pandas, matplotlib)
   - “re-bin” in order to have enough statistics for Eff
     - re-bin could be tuned



Code in:
effVsLumi.py

Conclusions and Plans

- Long-term stability monitoring development is on going
- DT Eff Vs int. Lumi plots for phi layers produced
- code ready for producing different outputs in order to cope with the trend monitoring strategy

Plans

- clean the code and put it on a github repository
- produce plots for all 2016 Runs and theta layers
- automatise the procedure creating a workflow

Thank you