Database for simulations

Pietro Meloni and Igor Abritta Costa
28-03-2023

How it works now

The user can either set a flag in the config file or run a script after the digitization.

In both cases, the code:

- - 1. uploads the output folder to the cloud **/cygno-sim/digitization-sql-registered**

- - 2. updates the sql table

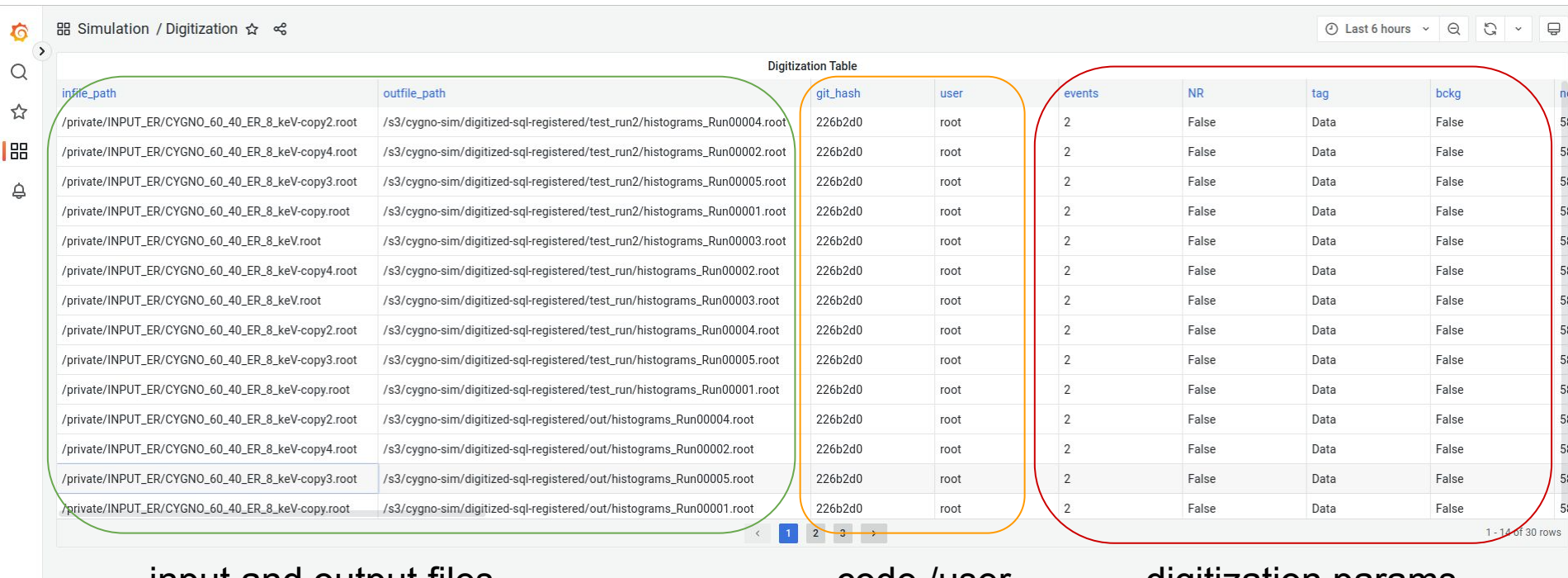   https://grafana.cygno.cloud.infn.it/d/BrJzxSBVk/digitization?orgId=1

Local files are not removed

If the user tries to upload twice the same output folder, it will overwrite it (and the table won't be updated)

How to access the table (with grafana)

https://grafana.cygno.cloud.infn.it/d/BrJzxSBVk/digitization?orgId=1

infile_path	outfile_path	git_hash	user	events	NR	tag	bckg
/private/INPUT_ER/CYGNO_60_40_ER_8_keV-copy2.root	/s3/cygno-sim/digitized-sql-registered/test_run2/histograms_Run00004.root	226b2d0	root	2	False	Data	False
/private/INPUT_ER/CYGNO_60_40_ER_8_keV-copy4.root	/s3/cygno-sim/digitized-sql-registered/test_run2/histograms_Run00002.root	226b2d0	root	2	False	Data	False
/private/INPUT_ER/CYGNO_60_40_ER_8_keV-copy3.root	/s3/cygno-sim/digitized-sql-registered/test_run2/histograms_Run00005.root	226b2d0	root	2	False	Data	False
/private/INPUT_ER/CYGNO_60_40_ER_8_keV-copy.root	/s3/cygno-sim/digitized-sql-registered/test_run2/histograms_Run00001.root	226b2d0	root	2	False	Data	False
/private/INPUT_ER/CYGNO_60_40_ER_8_keV.root	/s3/cygno-sim/digitized-sql-registered/test_run2/histograms_Run00003.root	226b2d0	root	2	False	Data	False
/private/INPUT_ER/CYGNO_60_40_ER_8_keV-copy4.root	/s3/cygno-sim/digitized-sql-registered/test_run/histograms_Run00002.root	226b2d0	root	2	False	Data	False
/private/INPUT_ER/CYGNO_60_40_ER_8_keV.root	/s3/cygno-sim/digitized-sql-registered/test_run/histograms_Run00003.root	226b2d0	root	2	False	Data	False
/private/INPUT_ER/CYGNO_60_40_ER_8_keV-copy2.root	/s3/cygno-sim/digitized-sql-registered/test_run/histograms_Run00004.root	226b2d0	root	2	False	Data	False
/private/INPUT_ER/CYGNO_60_40_ER_8_keV-copy3.root	/s3/cygno-sim/digitized-sql-registered/test_run/histograms_Run00005.root	226b2d0	root	2	False	Data	False
/private/INPUT_ER/CYGNO_60_40_ER_8_keV-copy.root	/s3/cygno-sim/digitized-sql-registered/test_run/histograms_Run00001.root	226b2d0	root	2	False	Data	False
/private/INPUT_ER/CYGNO_60_40_ER_8_keV-copy2.root	/s3/cygno-sim/digitized-sql-registered/out/histograms_Run00004.root	226b2d0	root	2	False	Data	False
/private/INPUT_ER/CYGNO_60_40_ER_8_keV-copy4.root	/s3/cygno-sim/digitized-sql-registered/out/histograms_Run00002.root	226b2d0	root	2	False	Data	False
/private/INPUT_ER/CYGNO_60_40_ER_8_keV-copy3.root	/s3/cygno-sim/digitized-sql-registered/out/histograms_Run00005.root	226b2d0	root	2	False	Data	False
/private/INPUT_ER/CYGNO_60_40_ER_8_keV-copy.root	/s3/cygno-sim/digitized-sql-registered/out/histograms_Run00001.root	226b2d0	root	2	False	Data	False

Digitization Table

1 - 14 of 30 rows

input and output files   code /user   digitization params

How to access the sql table (with pandas)

If the user wants to find specific digitized images that satisfie some conditions.

Example: The user wants the paths of all digitized images generated with these tracks:

/private/INPUT_ER/CYGNO_60_40_ER_8_keV.root

```
from sql2dataframe import *

df=sql2dataframe()

file_list=df[df["infile_path"]=="/private/INPUT_ER/CYGNO_60_40_ER_8_keV.root"]["outfile_path"].to_list()
for file in file_list:
    print(file)
```

```
/s3/cygno-sim/digitized-sql-registered/test_run2/histograms_Run00003.root
/s3/cygno-sim/digitized-sql-registered/test_run/histograms_Run00003.root
/s3/cygno-sim/digitized-sql-registered/out/histograms_Run00003.root
/s3/cygno-sim/digitized-sql-registered/run_test3/histograms_Run00003.root
/s3/cygno-sim/digitized-sql-registered/run_test4/histograms_Run00003.root
/s3/cygno-sim/digitized-sql-registered/test_run4/histograms_Run00003.root
```

How files look like in the cloud

```
cygno-sim/
└── digitized-sql-registered
    ├── LIME_24_keV_Sat_20_30cm_5790bgk_RitaDiff
    │   ├── histograms_Run00001.root
    │   ├── histograms_Run00002.root
    │   └── histograms_Run00003.root
    ├── LIME_Ca_keV_Sat_NewCode_Iso_RandZ
    │   └── histograms_Run00001.root
    ├── LIMEsaturation_1kVdrift_A1_beta1_sT0350_20cm_GEM1HV350_pedrun4159
    │   └── histograms_Run00001.root
    ├── LIMEsaturation_1kVdrift_A1_beta1_sT0350_30cm_GEM1HV350_pedrun4159
    │   └── histograms_Run00001.root
    ├── LIMEsaturation_1kVdrift_A1_beta1_sT0350_35cm_GEM1HV431_pedrun4159
    │   └── histograms_Run00001.root
    ├── LIMEsaturation_1kVdrift_A1_beta1_sT0350_45cm_GEM1HV440_pedrun4159
    │   ├── histograms_Run00001.root
    │   ├── histograms_Run00002.root
    │   ├── histograms_Run00003.root
    │   ├── histograms_Run00004.root
    │   └── histograms_Run00005.root
    └── LIME_Ti_keV_Sat_NewCode_Iso_RandZ
        ├── histograms_Run00001.root
        └── histograms_Run00002.root
```

Conclusions

- Ready to be added on github (**but not tested on condor…)**

- The script could be easily adapted for a track database (to run manually after G4/srim)

Notes:

- sometimes a connectivity error occurs while uploading files, in that case the table is not updated and the script will print "*An error has occurred!*" at the end fo output

- What happens if the user tries to upload the same folder? the files will be overwritten but the table won't be updated

- the script doesn't work on LNGS cluster or local PC (you can't upload files to the cloud)