technische universität
dortmund

MAGIC
Major Atmospheric
Gamma Imaging
Cerenkov Telescopes

The autoMAGIC Analysis Config File

autoMAGIC Workshop Padova

Jan Lukas Schubert

28th - 30th August 2024

Contents:

- The autoMAGIC analysis configuration file
  - Target specification
  - Analysis options
  - Common mistakes
- Setting up a configuration file for a test analysis

General Remarks

- You should create one autoMAGIC analysis config file for each analysis
  - Good means of documentation and important means for reproducibility!

- Before starting the analysis, you have to write the path to the analysis config file you want to use into your user config!

- Let's go through the options and create your test analysis config files alongside

- Create the file
  `/nfs/pic.es/user/your_username/analaysis_configs/Crab_test_analysis.toml`

- You can find a working analysis config file here:
  `/pnfs/pic.es/data/magic/Legacy/AUTOMAGIC/Crab_test_analysis.toml`

The autoMAGIC analysis configuration file

```
mars_version = "Mars-V3-1-0"

[target]
source_name = "CrabNebula"
#source_list_file = "list_of_sources.txt"        # path to file with one source name per line
start_date = 2018-11-01                          # use - as separator, not _ !
stop_date = 2019-09-15

[data_selection]
L1Table = "L1_3NN"
L3Table = ""                                     # take "L3T_SUMSUM_100_SYNC" for SumT data
transmission_9km_min_off = 0.8
transmission_9km_min = 0.55
transmission_9km_max = 1.2
zd_min = 5
zd_max = 62
dc_min = 0
dc_max = 2200
hv_setting = "NominalHV"
mola_threshold = 2
cloudiness_max_off = 20
cloudiness_max_on = 45
use_broken_lidar_data = true
calibrated_version_M1 = "current"                # use "current" for current version (highly recommended!) or specify version like "v1"
calibrated_version_M2 = "current"                # If a special version is required, make sure the version number is available at the chosen start_date!

[star]
cleaning_method = "sum"                          # Choose the cleaning algorithm you want to use. Default is "sum" cleaning. "mataju" cleaning is sometimes useful for lowE analyses
cl_lv1 = false                                   # User defined cleaning level 1. Taken from moon condition if not defined (highly recommended!)
cl_lv2 = false                                   # User defined cleaning level 1. Taken from moon condition if not defined (highly recommended!)
noise_lv_mean = false                            # User defined noise mean. Taken from moon condition if not defined (highly recommended!)
noise_lv_rms = false                             # User defined noise rms. Taken from moon condition if not defined (highly recommended!)
store_star_output = false                        # If true: Move star output files to /pnfs

[superstar]
ignore_mars_version_of_superstar_file = true

[RF]
cleaning_survival_rate = 0.5
force_rf = true
only_one_RF = false
max_underpopulated_bins_RF_check = 25
check_off_data_for_current_calibrated_version = true     # Check if RF is trained with the latest calibrated version
energy_estimation_method = "RFenStereo"                  # "LUTs" or "RFenStereo" decides if Look Up Tables or Random Forest is used for energy estimation in Melibea

[mc_parameters]
corsika_versions = ["mmcs699"]                   # "mmcs699", "mmcs6500", "std20130415", "std20140317", "std20141222", "std20150302"
view_cone = "ringwobble"                         # "ringwobble", "diffuse1.5", "diffuse2.5", diffuse4.0"
dont_care_view_cone = true                       # allow use of other MC with other view cones, if selected view_cone is not available
mc_trigger_type = "standard"                     # Other options: "SUMT", "EGALSUMT", "MATAJU" and "EGALMATAJU"
extension = ""                                   # If you want to use a highE or lowE energy extension here, use "highE" or "lowE"
ignore_mars_version_of_selectmc_file = false

[melibea]
use_lidar_correction = true
forced_coach_job_ids = false                     # If false: use/create matching coach job (recommendend!), if [1691, 1693]: use corresponding coach jobs (e.g. for Crab Check)
force_diffuse_mcs = false

[magicDL3]
dl3_converter_version = "v0.1.15"
irf_type = "point-like"                          # "point-like", "full-enclosure"
az_bins = 1
hadronness_cut_from_efficiency = true            # optimize hadronness cut energy bin-wise as quantile of surviving MC events, default: true
theta2_cut_from_efficiency = true                # theta2 cut energy bin-wise as quantile of MC events, default: false (Gammapy does currently not work with bin-wise cuts!)
hadronness_cut = 0.3                             # events with hadronness < cut will be selected as gamma events; used if hadronness_cut_from_efficiency = false
theta2_cut = 0.02                                # MC events with theta2 < cut will be selected for computing the IRFs; used if theta2_cut_from_efficiency = false
quantile_hadronness_cut = 0.9                    # quantile of MC events to survive the hadronness_cut (applied energy bin-wise)
quantile_theta2_cut = 0.9                        # quantile of MC events to survive the theta2 cut (applied energy bin-wise)
```

Target specification

- Source list: Used for massive productions (special use case)

```
1   mars_version = "Mars-V3-1-0"
2
3   [target]
4   source_name = "CrabNebula"
5   #source_list_file = "list_of_sources.txt"       # path to file with one source name per line
6   start_date = 2018-11-01                         # use - as separator, not _ !
7   stop_date = 2019-09-15
```

Analysis Cuts

- Take Care with the L1 and L3 Table!
  - If you mess it up you will use entirely different data types and the analysis might be worthless!
- Hazy atmosphere conditions not implemented in DL3 converter/gammapy
  - Use higher transmission cuts

```
[data_selection]
L1Table = "L1_3NN"
L3Table = ""                                # take "L3T_SUMSUM_100_SYNC" for SumT data
transmission_9km_min_off = 0.8
transmission_9km_min = 0.55
transmission_9km_max = 1.2
zd_min = 5
zd_max = 62
dc_min = 0
dc_max = 2200
hv_setting = "NominalHV"
mola_threshold = 2
cloudiness_max_off = 20
cloudiness_max_on = 45
use_broken_lidar_data = true
calibrated_version_M1 = "current"           # use "current" for current version (highly recommended!) or specify version like "v1"
calibrated_version_M2 = "current"           # If a special version is required, make sure the version number is available at the chosen start_date!
```

Analysis Options – star/superstar

- Take care to use the right cleaning method here!
  - Standard: Sum cleaning
- Only set cleaning levels here if you don't want to use the standard ones!
- Ignore mars for superstar is almost always recommended!

```
27  [star]
28  cleaning_method = "sum"                    # Choose the cleaning algorithm you want to use. Default is "sum" cleaning. "mataju" cleaning is sometimes useful
29  cl_lv1 = false                             # User defined cleaning level 1. Taken from moon condition if not defined (highly recommended!)
30  cl_lv2 = false                             # User defined cleaning level 1. Taken from moon condition if not defined (highly recommended!)
31  noise_lv_mean = false                      # User defined noise mean. Taken from moon condition if not defined (highly recommended!)
32  noise_lv_rms = false                       # User defined noise rms. Taken from moon condition if not defined (highly recommended!)
33  store_star_output = false                  # If true: Move star output files to /pnfs
34
35  [superstar]
36  ignore_mars_version_of_superstar_file = true
```

Analysis Options – coach

- force_rf is always recommended!
- The RF for the energy estimation if preferred over the LUTs
- only_one_RF should only be enabled in case of little OFF data!

```
38  [RF]
39  cleaning_survival_rate = 0.5
40  force_rf = true
41  only_one_RF = false
42  max_underpopulated_bins_RF_check = 25
43  check_off_data_for_current_calibrated_version = true    # Check if RF is trained with the latest calibrated version
44  energy_estimation_method = "RFenStereo"                 # "LUTs" or "RFenStereo" decides if Look Up Tables or Random Forest is used for energy estimation in Melibea
```

Analysis Options – MCs

- You can use multiple CORSIKA versions in parallel to increase the statistics
- For extended sources use diffuse MCs!
- Except the source is very huge, dont_care_view_cone is recommended
- Extensions only needed in special analysis cases
- Ignore mars makes sense for analyses starting on superstar data

```
46  [mc_parameters]
47  corsika_versions = ["mmcs699"]                  # "mmcs699", "mmcs6500", "std20130415", "std20140317", "std20141222", "std20150302"
48  view_cone = "ringwobble"                        # "ringwobble", "diffuse1.5", "diffuse2.5", diffuse4.0"
49  dont_care_view_cone = true                      # allow use of other MC with other view cones, if selected view_cone is not available
50  mc_trigger_type = "standard"                    # Other options: "SUMT", "EGALSUMT", "MATAJU" and "EGALMATAJU"
51  extension = ""                                  # If you want to use a highE or lowE energy extension here, use "highE" or "lowE"
52  ignore_mars_version_of_selectmc_file = false
```

Analysis Options – melibea

- forced_coach_job_ids can make sense for crab checks!
- Not using broken lidar data will discard much data but only use data in which we know the atmospheric conditions well

```
54  [melibea]
55  use_lidar_correction = true
56  forced_coach_job_ids = false              # If false: use/create matching coach job (recommendend!), if [1691, 1693]: use corresponding coach jobs
57  force_diffuse_mcs = false
```

Analysis Options – DL3 Converter

- Standard case: point like IRFs
- Hadronness and theta^2 cuts from efficiency recommended
  - Energy dependent cuts possible
- Quantiles will override hard cuts in case cuts from efficiency is enabled

```
[magicDL3]
dl3_converter_version = "v0.1.15"
irf_type = "point-like"                    # "point-like", "full-enclosure"
az_bins = 1
hadronness_cut_from_efficiency = true      # optimize hadronness cut energy bin-wise as quantile of surviving MC events, default: true
theta2_cut_from_efficiency = true          # theta2 cut energy bin-wise as quantile of MC events, default: false (Gammapy does currently not work with bin-wise
hadronness_cut = 0.3                       # events with hadronness < cut will be selected as gamma events; used if hadronness_cut_from_efficiency = false
theta2_cut = 0.02                          # MC events with theta2 < cut will be selected for computing the IRFs; used if theta2_cut_from_efficiency = false
quantile_hadronness_cut = 0.9              # quantile of MC events to survive the hadronness_cut (applied energy bin-wise)
quantile_theta2_cut = 0.9                  # quantile of MC events to survive the theta2 cut (applied energy bin-wise)
```