Real Time Analysis - kick off meeting

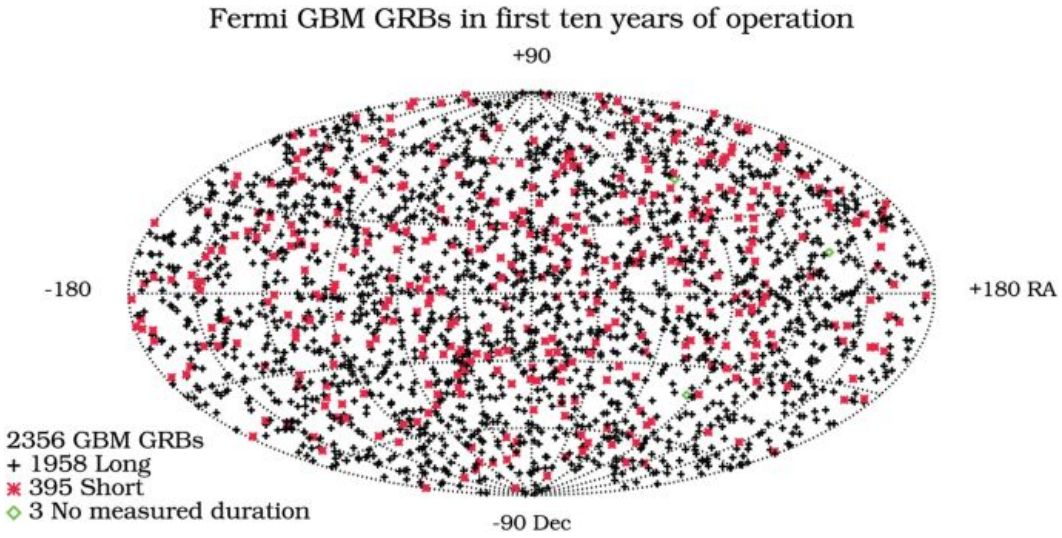
Sami Caroff, Pierre Aubert, Vincent Pollet

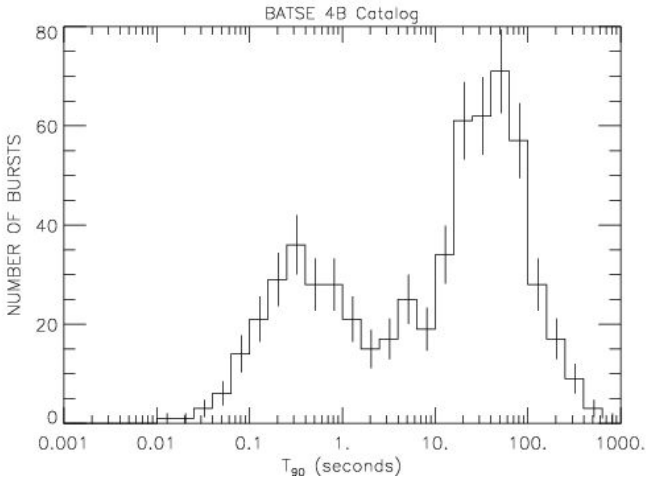
06/11/2023

Funded by the European Unions's Horizon 2020 – Grant N˚824064

Why do we need a Real Time analysis ?

- Major Class of gamma ray sources are transient (GRBs, NSNS merger, FRBs…) or variable sources (AGNs)
- Transient : can happen anytime, anywhere in the sky (extragalactic sources)
- Variable source : can happen anytime, but catalog of source can help to know where to look (catalog of known sources)
- Problem : Cherenkov telescope have a tiny field of view (~5°)





Satellite versus ground observatory

Fermi telescope (satellite)

- 1 m^2 (low surface)
- Big FoV (~ half of the sky)
- ~Always work (don't care of day/night cycle, weather)
- angular resolution ~few degrees

Cherenkov telescope

- 0.1 km^2 (big surface)
- small FoV (~ 5 deg)
- ~Never work (work by night, without moon, good weather)
- angular resolution ~0.2°

- Complementarity between satellites and ground observatory
- Satellite use their big FoV and their high duty time to provide alert to ground observatory
- Cherenkov Telescope follow if they can…

LHAASO

18 TeV
100 sigma
5000 photons

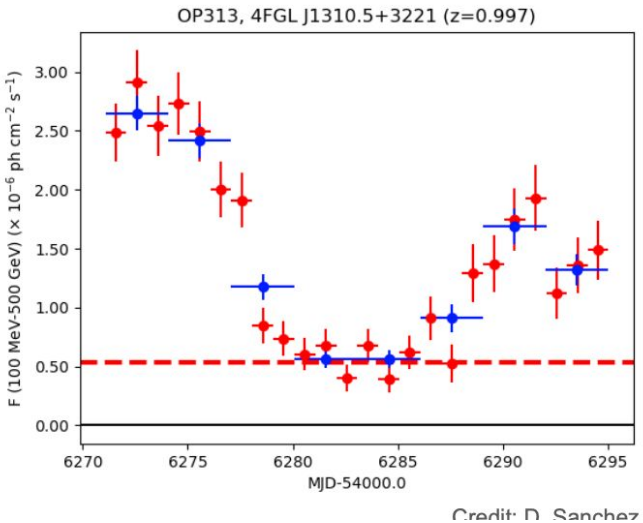
Credit: S.Zhu/D.Green

IACTs

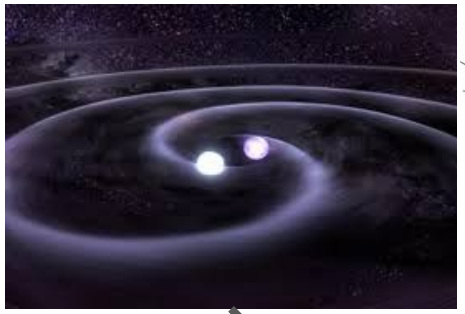
the mooooon
is too briiiiiiight

Following alerts

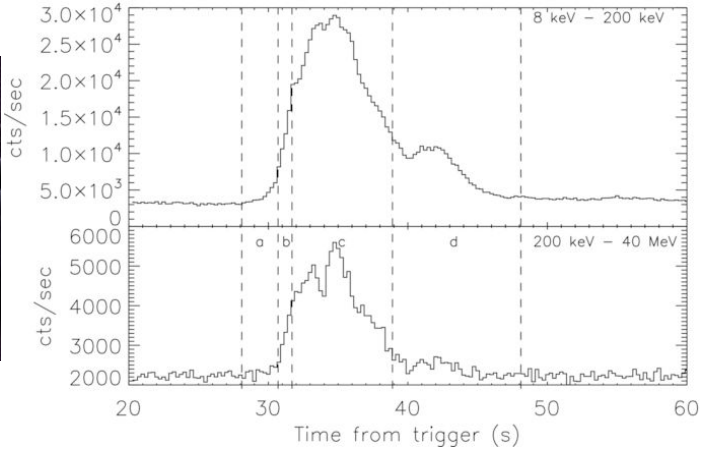
AGN flaring



GW alert



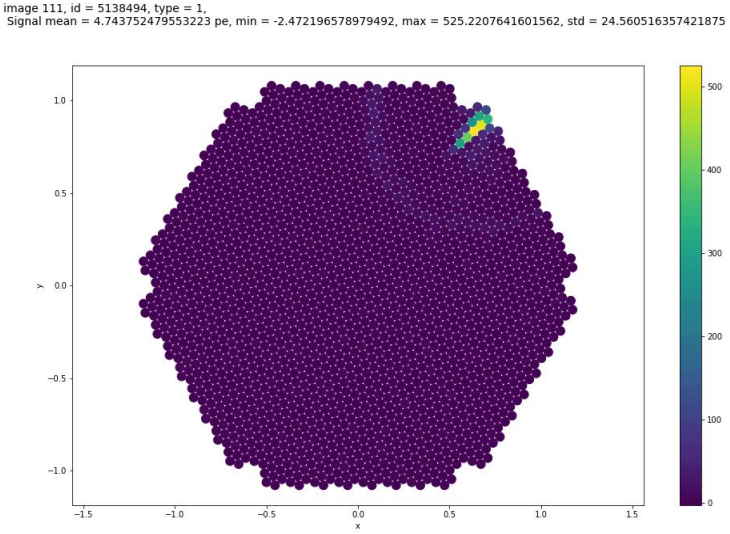
GRB detected



- Before receiving an alert, LST already had a schedule of sources to observe
- Following of an alert need to discard planned observation (because tiny field of view !)
- Need to optimise how much time do we observe (to not waste time on undetectable object)
- Need to know where to look (not precise position of the alert on the sky)

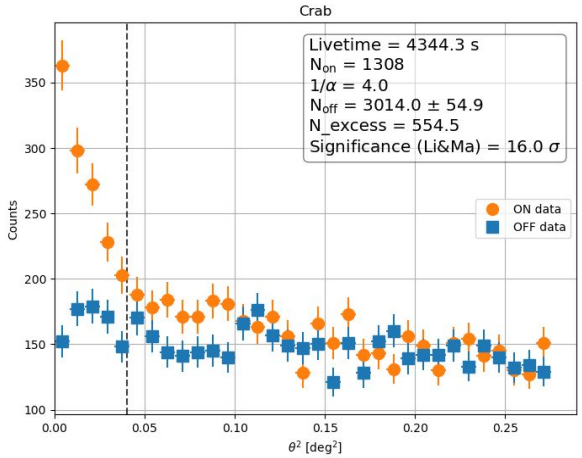
Real Time Analysis

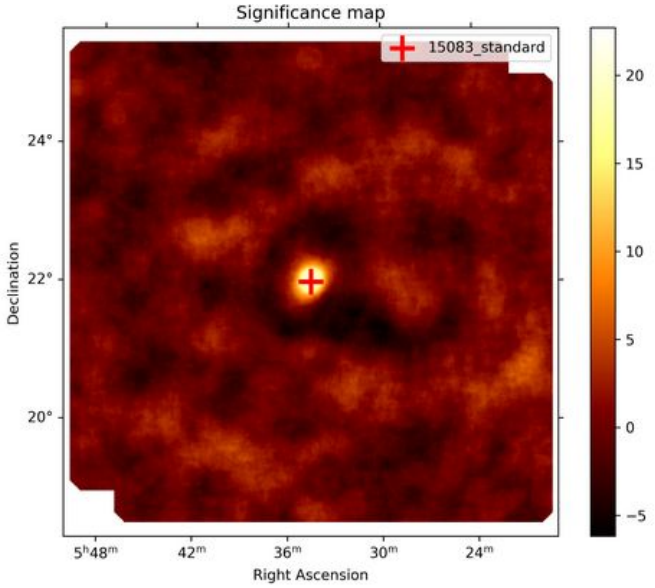
Without real time analysis



Good look to find a source and understand what you are looking

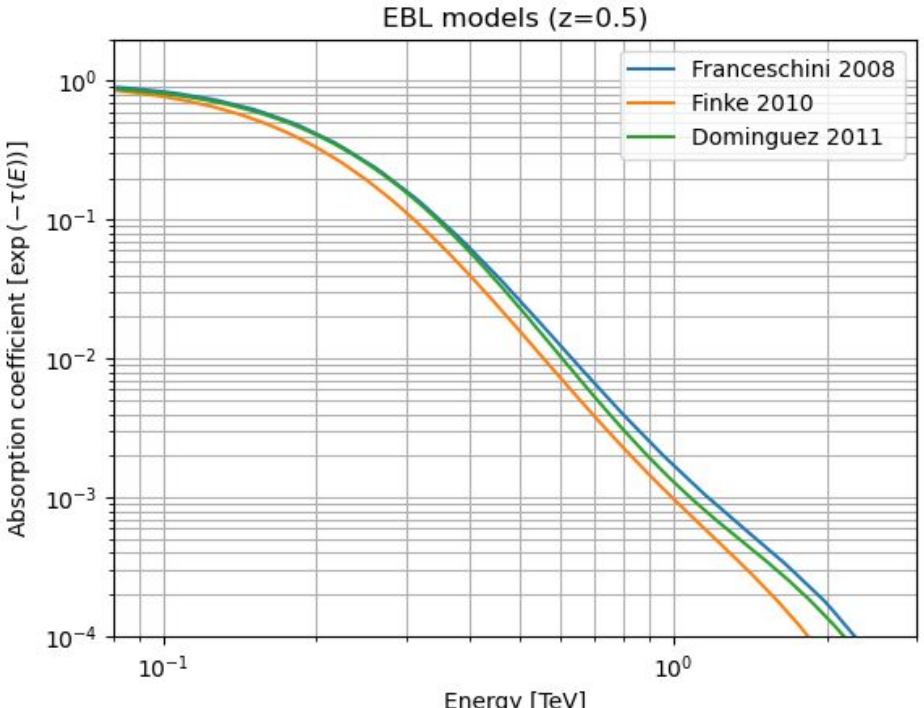
With real time analysis





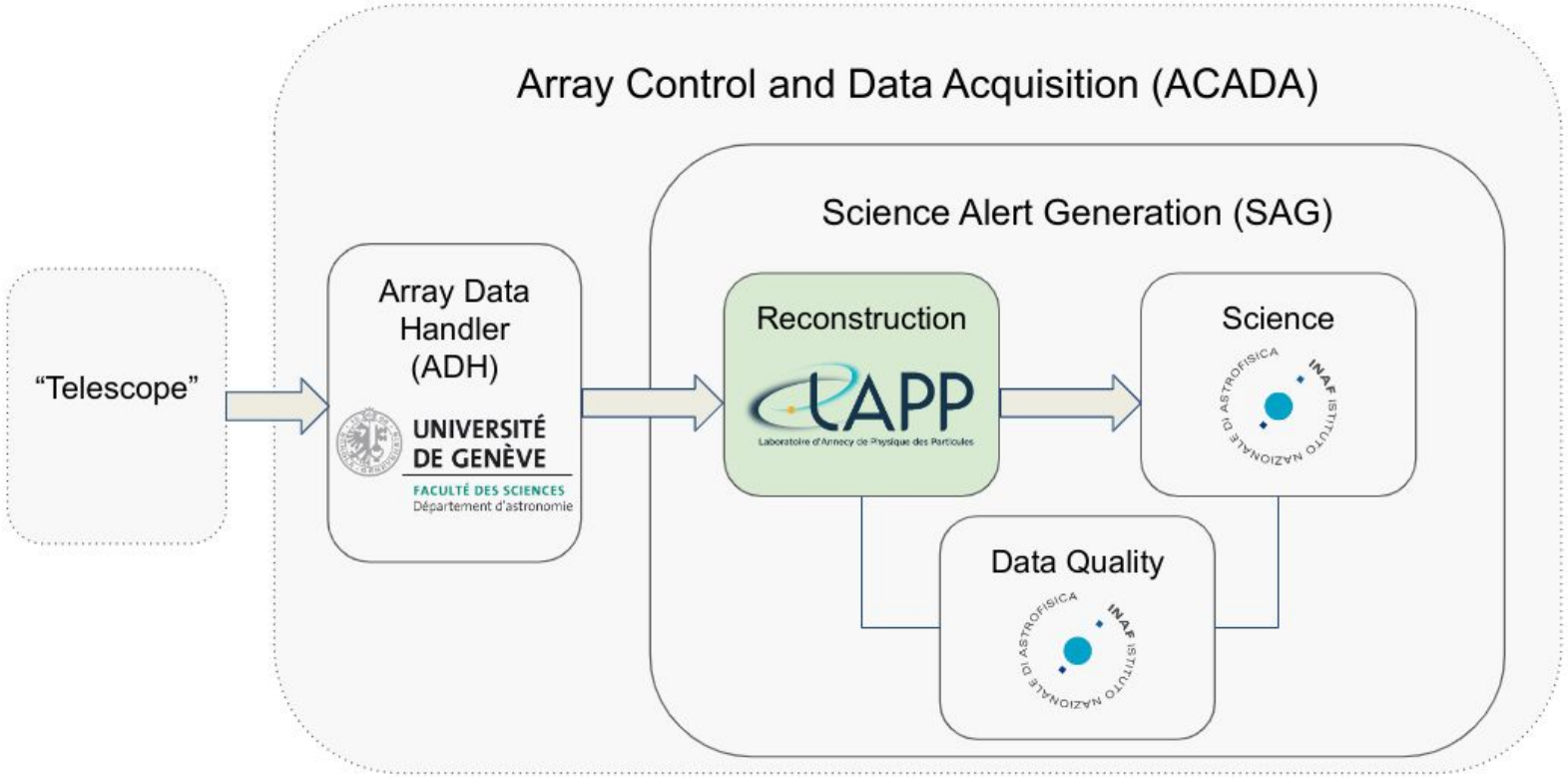
Real Time Analysis

- RTA need to be fast to give fast feedback to (15 sec to produce DL3)
- Efficient (at least 50% as efficient than requirement of offline analysis)
- We need to have a low energy threshold :
  - because most of the interesting sources for RTA are extragalactic, and high energy gamma ray are absorbed by Extragalactic Background Light
- The lowest we go in energy, the further we look (for RTA further source detected is z ~ 0.45 so far -> ~4 milliards d'année lumière)
- Furthest source observed by RTA but not detected z=1 (7 milliards d'année lumière)
- But the lowest we go in energy, the more difficult it is to reconstruct and discriminate events



Real-time analysis - ACADA-SAG

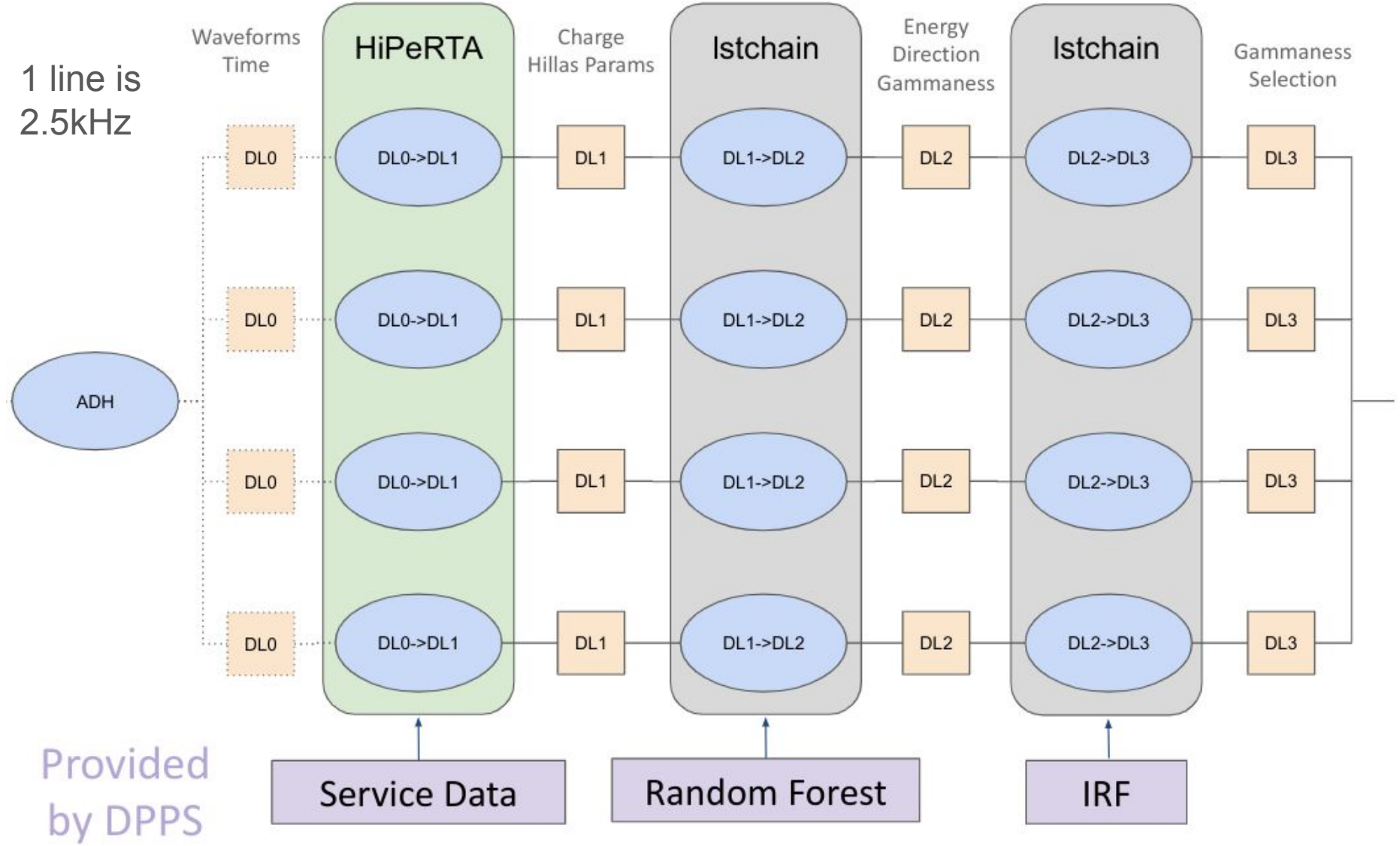
- Real-time analysis objectives :
  - **Real-time production of DL3 data (LAPP)**
  - Real-time Data quality
  - Real-time High-level analysis for science monitoring (DL4 DL5)
  - Real-time generation of candidate science alerts
- Currently developing and testing this framework on LST-1 (LAPP, INAF)



Real time analysis pipeline

https://gitlab.in2p3.fr/CTA-LAPP/HiPeRTA

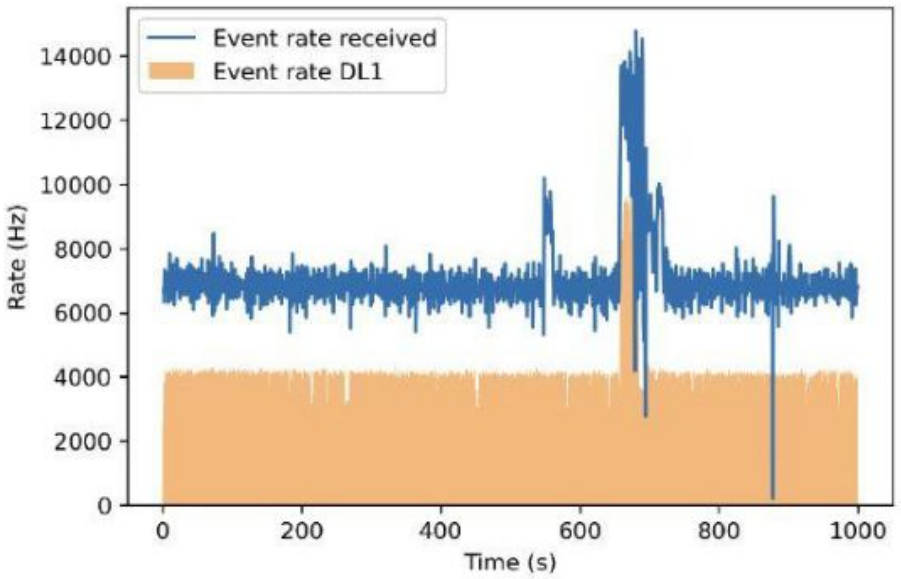
https://github.com/cta-observatory/cta-lstchain

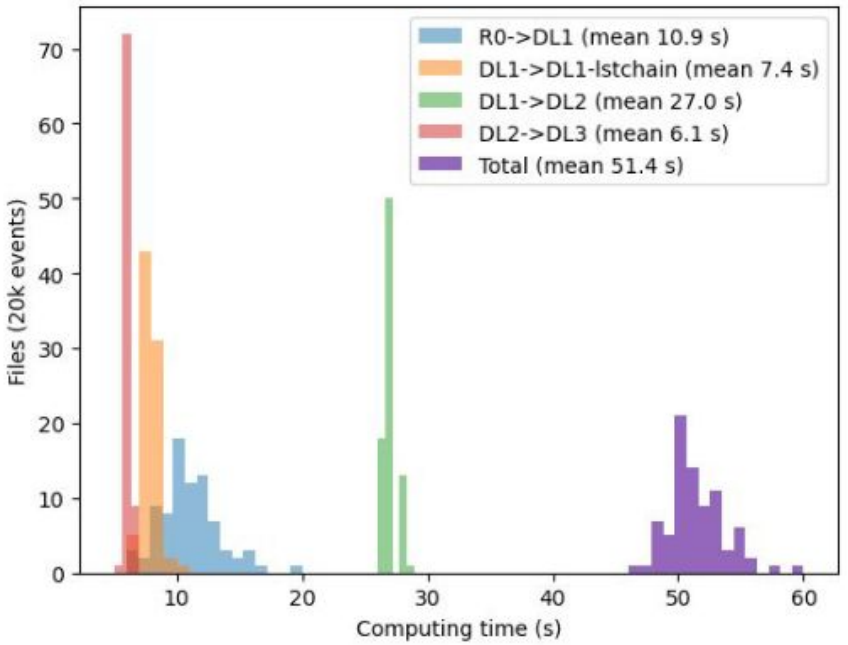


Automation of the pipeline

- Pipeline triggered by a database filled during night by the Telescope Control Unit (TCU), with a supervisor code (https://gitlab.in2p3.fr/CTA-LAPP/rta/lst_auto_rta)
- LST Auto RTA is made to be the interface between the orchestrator code and the TCU DB (basically it write a config based on the information in the database)
- Orchestrator python code (https://gitlab.in2p3.fr/CTA-LAPP/rta/HiPeRTA_Stream) in order to send jobs to slurm based on a configuration
- 4 machines with 32 cores reserved by night for RTA, using slurm to schedule jobs

Performance (Computing time)

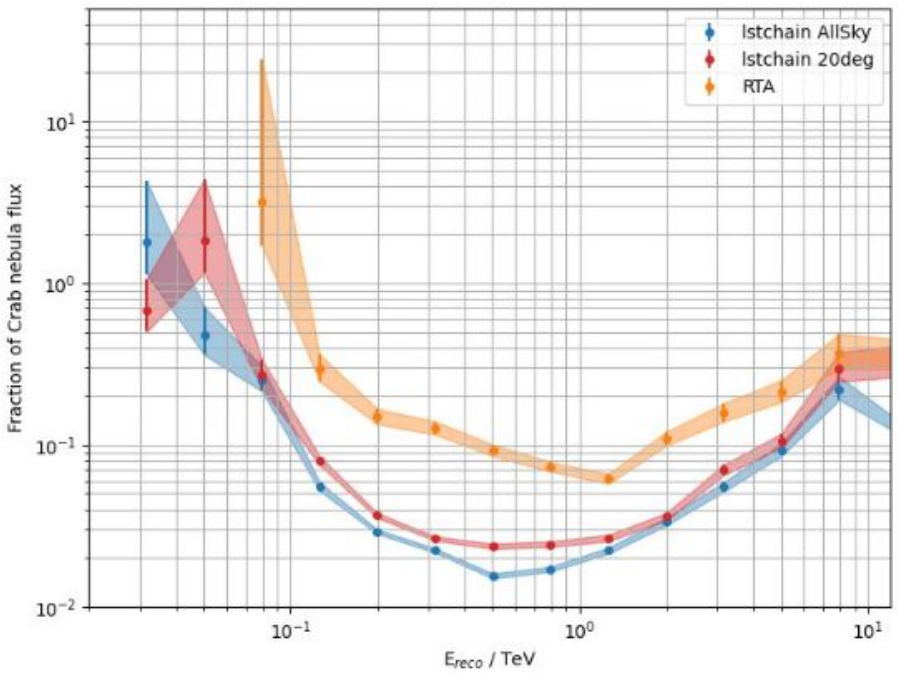


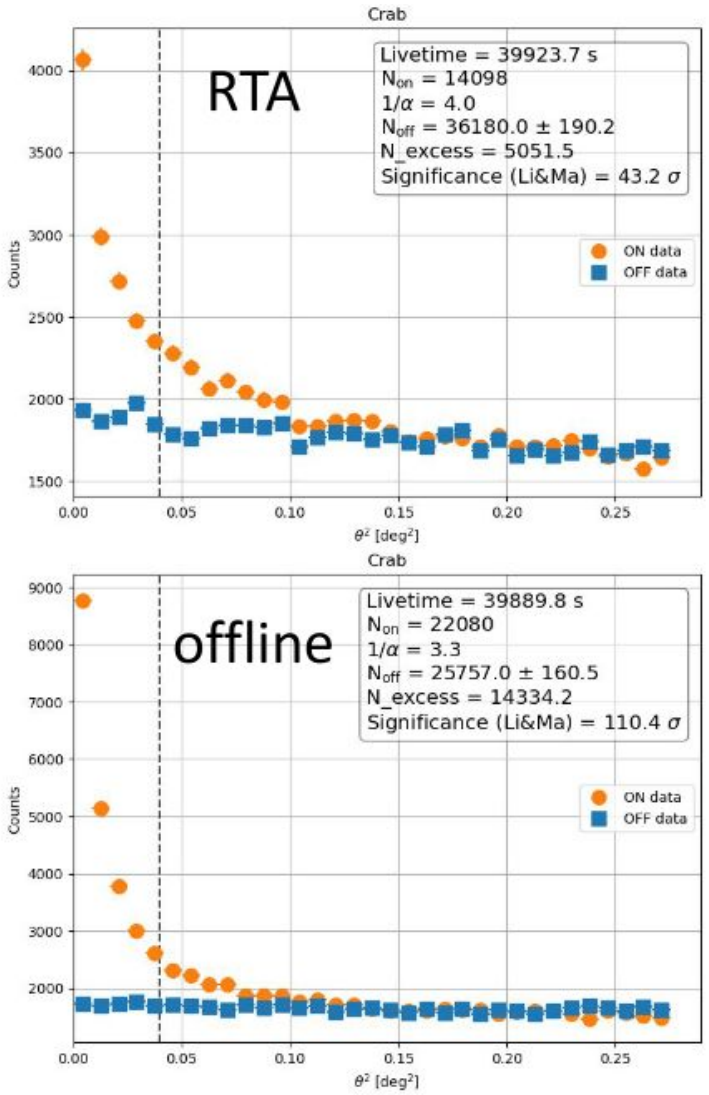


- Current configuration:
  - one new file every 20k events
  - 4 streams * 1 cores used
  - rate per core ~ 2000Hz

- Speed requirements: Total time < 15 s
- Speed comparison R0->DL1:
  - offline analysis 30 ms/event
  - RTA ~0.5 ms/event
- Next steps:
  - Remove conversion step
  - optimize DL1->DL2, DL2->DL3

Reconstruction performance

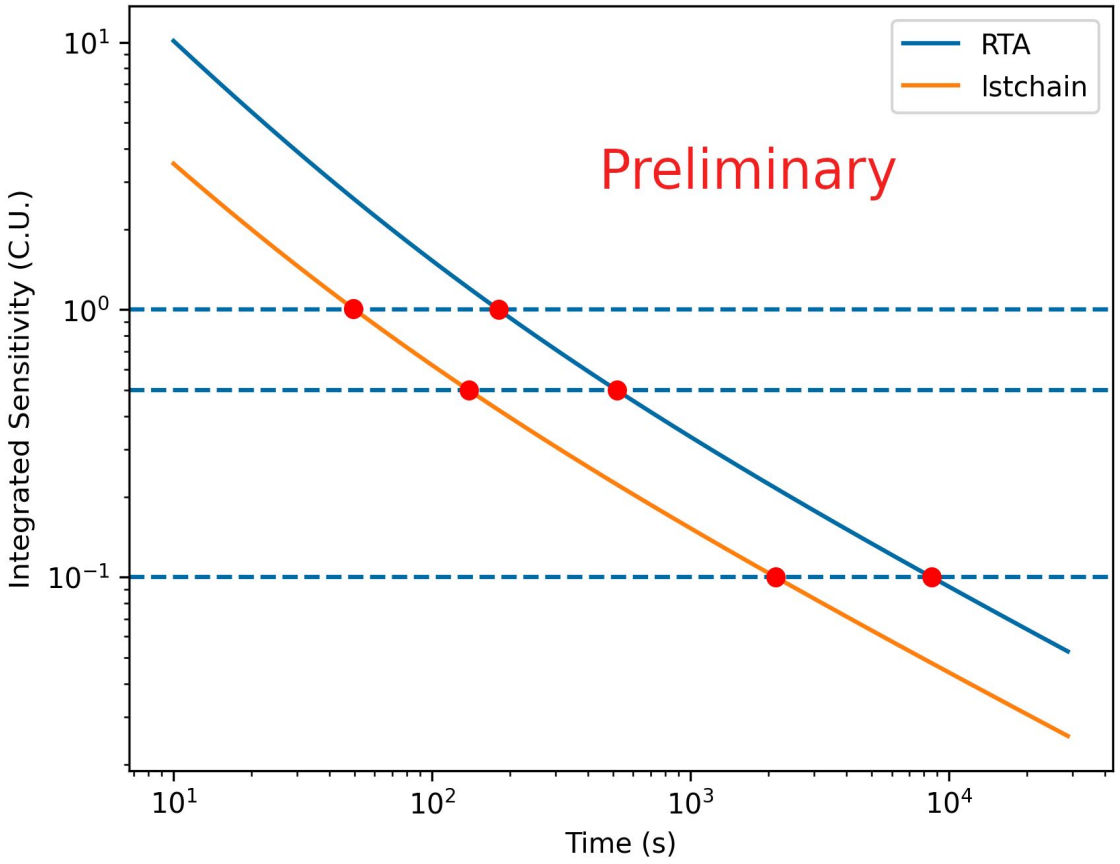




- Accumulated Crab data since December 2023
  - Benchmarking data set for RTA improvements

Performance (Mkn 421)

- Sensitivity extracted from Mkn 421 data optimised in gammaness (not per bin) for both chain
- Sensitivity around lstchain * 2 above 0.4 TeV, worst below
- Crab detectable in ~3 minutes (in real-time test Mkn421 at ~1CU observed in 4 minutes)
- 0.3 Crab can be observed in a single run
- Performance obtained with Mkn 421 FoV, using lstchain to assess the Mkn421 flux for this particular observations -> not optimal, Crab data is safer



Why Gamma learn can be interesting for RTA ?

Pros

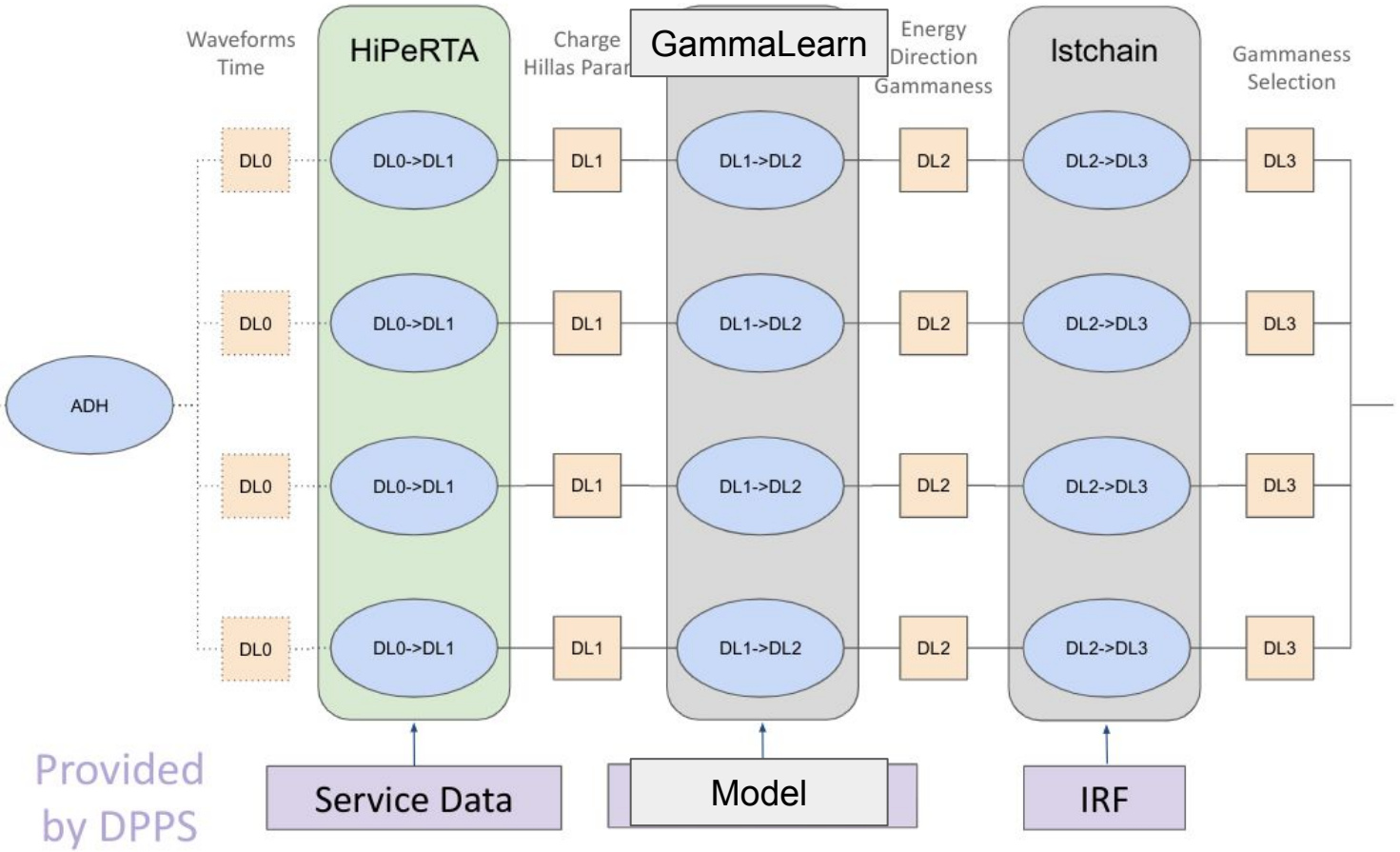
- Better performance at low energy (of course)
- DANN method can help MC/Data discrepancy (Cat A calibration, no run-wise simulation possible in real time)

Cons

- MC/Data discrepancy is by definition higher in RTA than in offline analysis (possible problem to use Gamma learn Vanilla)

Crossover GammaLearn X RTA

- RTA is a modular software, DL1->DL2 step can be done by an other software (if input and output are preserved)
- Can even be done be done in parallel since it is not using competing ressources (CPU vs GPUs) (apart for the DL2->DL3 steps)



Tasks and discussion

- Integration of GammaLearn in the current framework
  - No need to modify anything in RTA (DL1 already produced with image and time)
  - Mostly configuration issue for the scheduler
  - Need to have GPU reserved by night
  - Training data are already available (RTA DL1 MC available)
  - Optimisation need in Gammalearn to be real time ?
- Several benchmark can be used for test
  - DL1 Crab data
  - Stream data (to perform test by day with registered data)
- Performance ?
  - is Gamma learn vanilla good enough ?
  - How to deal with the issue of mismatch between models and reality
  - Can DANN be used, and if yes, what training sample should be used ?
  - Probably the most complicate part of the project
- Some other practic thoughts :
  - new GammaLearn model should be computed automatically for each new RTA version (integration of gammaLearn in lstmcpipe ?)
- Some other ideas :
  - Already some dev of a GPU R0->DL1, can we make a GPU R0->DL2 (Hiperta + gammalearn) in order to use more efficiently GPUs ?