Event reconstruction

from detector simulation to IRFs

M.Peresano, R. Covarelli, A. Negro

Università degli Studi di Torino & INFN Torino

SWGO Italia meeting - Naples - 24/11/23

nPE vs. r template production (1)

1. produce *unsmoothed* templates (daqsim & config files from the config-swgo)

```
lookuptable-generator --input /path/to/input.xcd --output-root-file
/path/to/output.root --config-file /path/to/config_A1.xml --daqsim -m
/path/to/daqsim-config_white.ini --make-UC-templates
--make-ABCD-templates
```

2. Get the mean and sigma for a specific r in each (Xmax, E, θ) bin with this file
3. re-run with `--input-root-file /path/to/mean_sigma_hist_of_logprob_profiles.root` to get the new layout of templates to prepare for smoothing
4. Smooth (Gaussian distributed weighted sum) in x- and y- direction of the templates (see README)

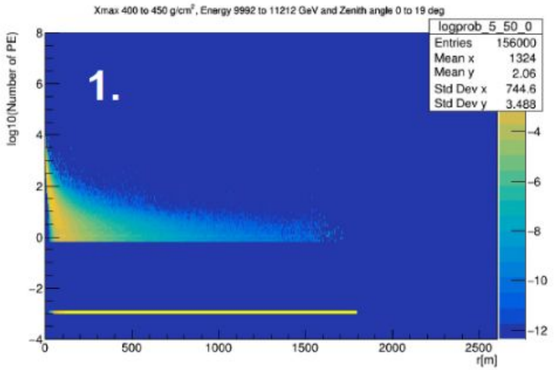
nPE vs. r template production (2)

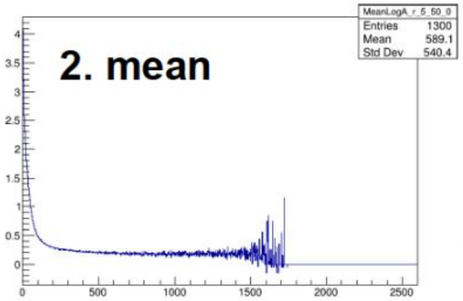
Templates used for reconstruction with `LHLatDistFit` module in AERIE

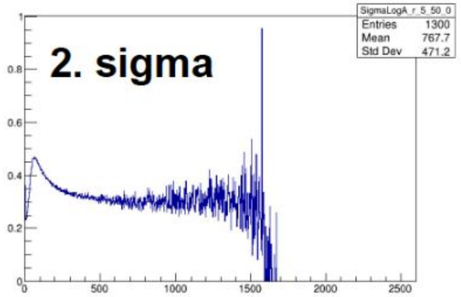
Energy template (N_hit vs Core distance) and Xmax vs energy are basically just single plots

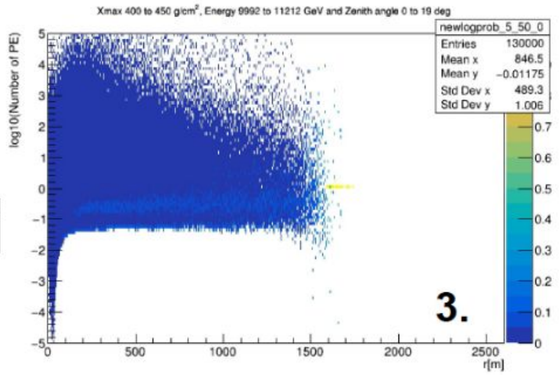
see production for M5 simulations

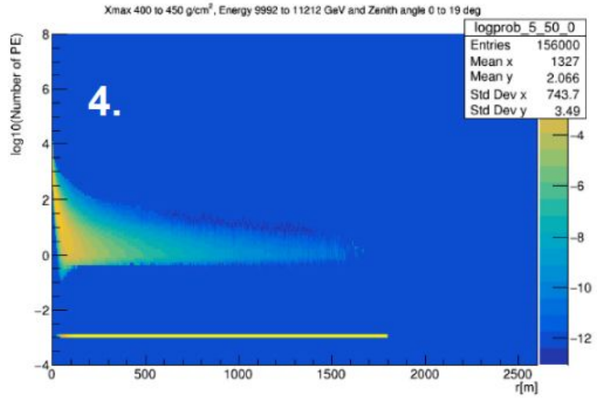
```
Look-up table generator input parameters:
  --output-root-file arg (~/lfs/l2/hawc/users/vikasj78/software/MyAERIE/model_out.root)
                                root file for output of core fitter,
                                ole
  --input-root-file arg (=Mean_RMS_file.root)
                                root file contianing the mean and sigma
                                information of the probability profiles
                                after the first iteration, ole
  --make-ABCD-templates         Add to make templates for tanks A, B,
                                C, and D
  --make-E-templates            Add to make templates for tank E
  --make-F-templates            Add to make templates for tank F
  --make-UC-templates           Add to make the upper cell templates
  --make-LC-templates           Add to make the lower cell templates
```











Example template creation for a (X_{max}, E, θ) bin for the A1 configuration

Reconstruction steps (1)

1. center of mass (COM) core guess → **first core estimate**
2. upload of **shower front curvature model**
3. 1st fit of the shower front with a plane model + application of curvature correction → **first shower direction estimate**

Reconstruction steps (2)

1. **Likelihood-based template fitting** reconstruction (LHLatDistFit module)
   a. (optional for E tanks) export PMT signal time trace
   b. use the core estimated previously as seed for the fit to get a more precise estimation
   c. use better core position to re-estimate shower direction
2. Fitting needs **5 .root files**:
   a. the upper cell LUT, one for each (xmax, E, theta) bin, each looks like plot 4
   b. lower cell LUT (not implemented yet - set equal to upper cell)
   c. energy = N_hit vs core distance
   d. xmax vs energy
   e. some placeholder file that would be used for a sort of Gooodness of Fit (not implemented yet)

Reconstruction steps (3)

5. get number of PEs produced by **muons** (needed for classification)
6. Estimate **classification** using many different single algorithms
    a. HAWC algorithms (see http://arxiv.org/abs/2205.12188)
        i. compactness
        ii. PINCness
        iii. Chi2 fit of lateral distribution function (LDFchi2)
    b. LCm (https://iopscience.iop.org/article/10.1088/1475-7516/2022/10/086)

Launching swgo-reco I

- **Necessary** software:
  - **AERIE** for swgo-reco
  - **config-swgo** to give swgo-reco all the needed configuration files
- **Preparation steps** (do once after downloading config-swgo):
  - `export CONFIG_SWGO=PATH/TO/config-swgo`
  - `$CONFIG_SWGO/M5/bin/m5-xml-maker`

m5-config-emitter

- swgo-reco requires **many arguments**
- They depend on array configuration
- `$CONFIG_SWGO/M5/bin/m5-config-emitter -$COLOR -c $M5CONFIG`

  returns a really long string that can be **directly passed to swgo-reco** command
- 2 arguments:
  - `$COLOR`: material of the upper layer (**b** for Polypropylene and **w** for Tyvek)
  - `$M5CONFIG`: the configuration identifier, must be **within the ones simulated in M5**

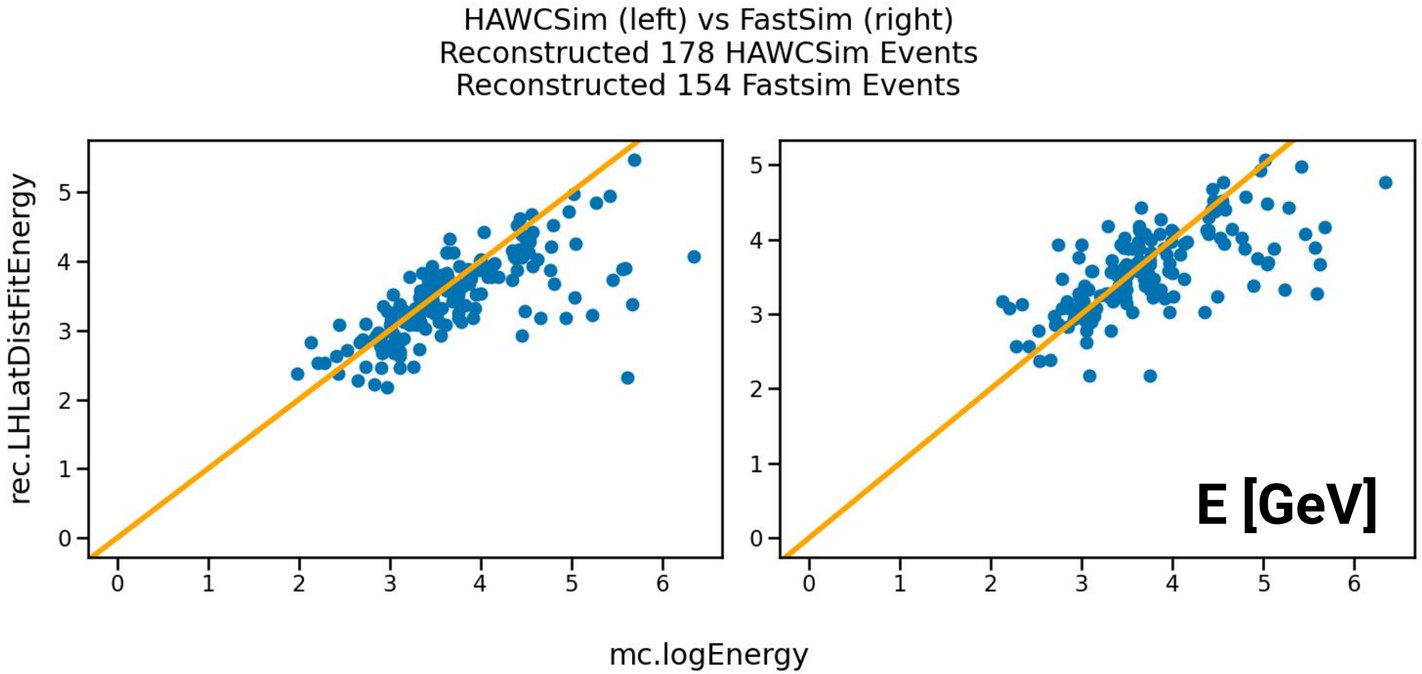
Launching swgo-reco II

1. `M5CONFIG=A2 # or any other template you are studying`
2. `COLOR=w # b is for black and w is for white`
3. `INPUT_FILE=PATH/TO/hawcsim_file.xcd`
4. `OUTPUT_FILE=PATH/TO/output_file.xcd`
5. ```
   swgo-reco `$CONFIG_SWGO/M5/bin/m5-config-emitter -$COLOR -c
   $M5CONFIG` --input $INPUT_FILE --output $OUTPUT_FILE
   ```

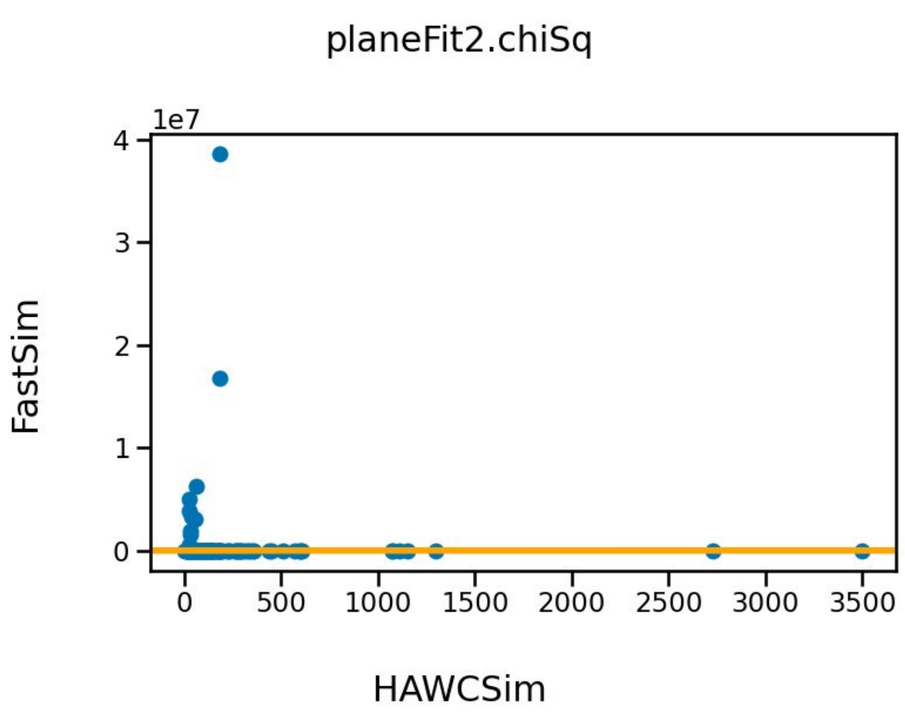
WARNING! The program appears to be demanding in resources, tests on normal PCs resulted in crashes while at CNAF swgo-reco runs
   ```

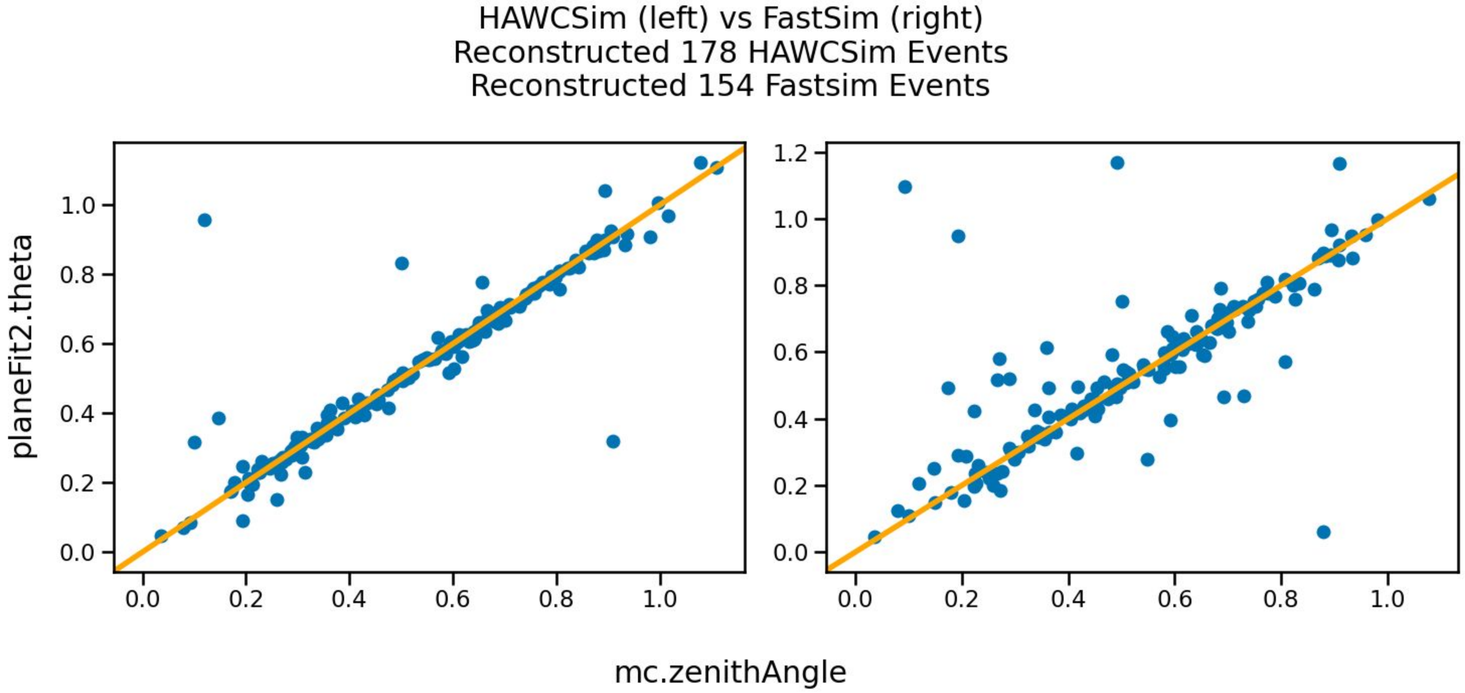
Fast Simulation and swgo-reco

- Tested on a version of fast simulation with XCDF output
- Checks done on a single proton file simulated for A2 array
- Controls between MC values / Fast Simulation / HAWCsim



Fast Simulation and swgo-reco



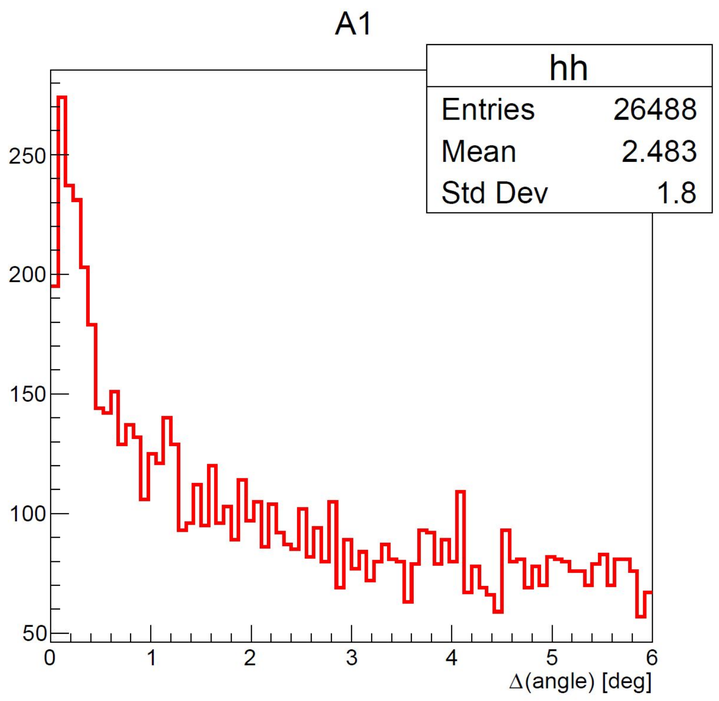


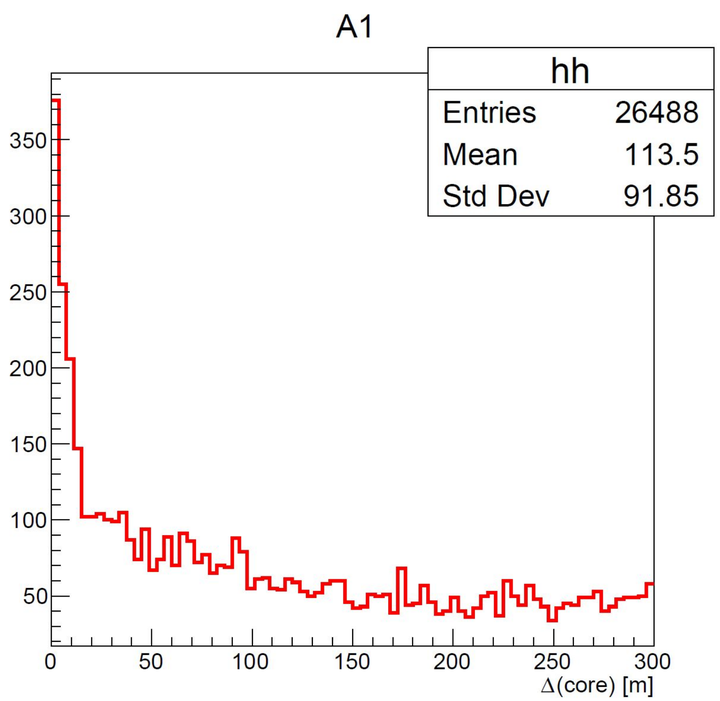
Some things still need to be understood

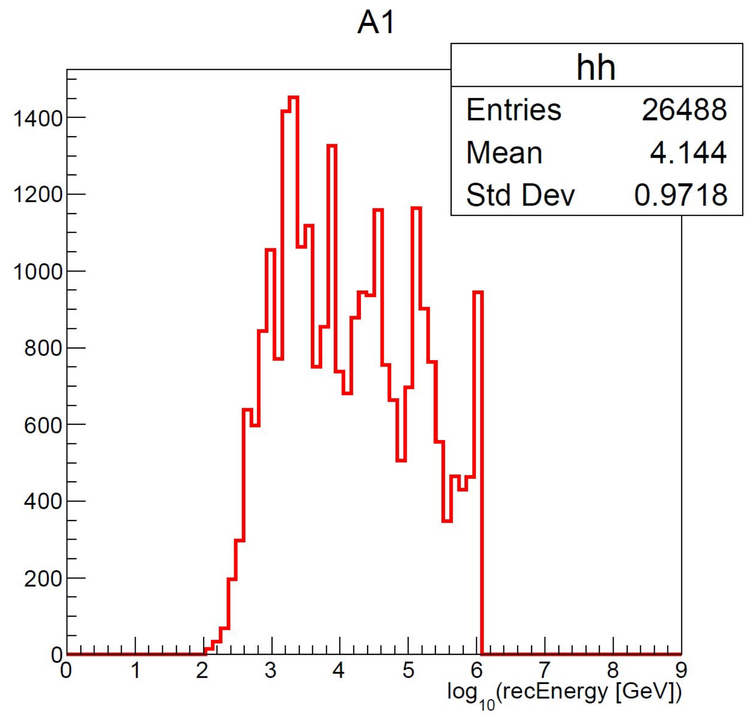
UHE events and swgo-reco

- Tried full `HAWCSim` + `swgo-reco` on CORSIKA production at CNAF (100 TeV- 10 PeV gammas and protons)
  - Workflow easily adapted to `HTCondor` logic
  - Very successful (almost 100% jobs done)
  - Negligible `swgo-reco` running time (minutes/event) w.r.t. `HAWCSim` (average 1h30/event, with large fluctuations strongly dependent on energy and Rcore)
  - NOTICE: to run on multiple files one can:
    - merge before using `xcdf` → NOT WORKING, resulting files corrupted
    - provide an input list to `swgo-reco` → OK

UHE events and swgo-reco







First steps ~OK, clear problem with template method (peaks + no templates available beyond 1 PeV?)

Reconstruction steps (4)

This part is performed from pyswgo.

1. from `swgo-reco` we obtain a set of .xcd files that can be converted to .root
2. `pyswgo-make-event-level` reduces data for training/estimation of final gammaness
3. `pyswgo-classify-events` processes output from step 2, produces set of cut files, and output file in which all gamma-hadron separators (slide 8) are combined in a single variable by means of an MPL classifier

Special case: Mercedes-like tanks (e.g. E tank design), see documentation.

Instrument Response Functions (IRFs)

For an analytical explanation see here.

Current implementation from HAWC (see docs):

1. survived reco events binned in (E, Θ, r) for M analysis bins each with a subset of “fHit” bins (aka N_hit_tanks)
2. all IRFs are map datasets (1 bin in Ra, N bins DEC, 1 or more non-spatial axes)
3. each DEC row a possible transit = zenith has been integrated
4. MR!88 computes sensitivity based on this input

Instrument Response Functions (IRFs)

Recent developments:

- we want to use IRFs in table format (i.e zenith still binned)
  - the Transients WG needs to remove the information of “a” simulated transit
  - from tables we can get easily maps via zenith integration in each DEC bin

- ⇒ we can drop old maps computation
- WIP: PSF under review, BKG2D coming soon (testing)

Data format follows GADF (frozen) and aims at VODF.