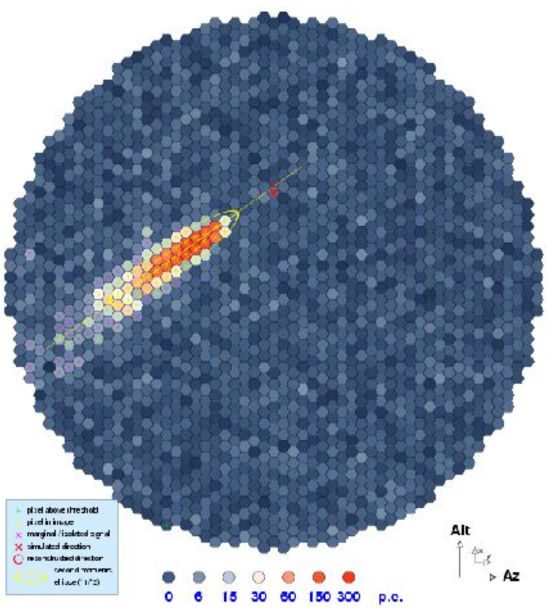
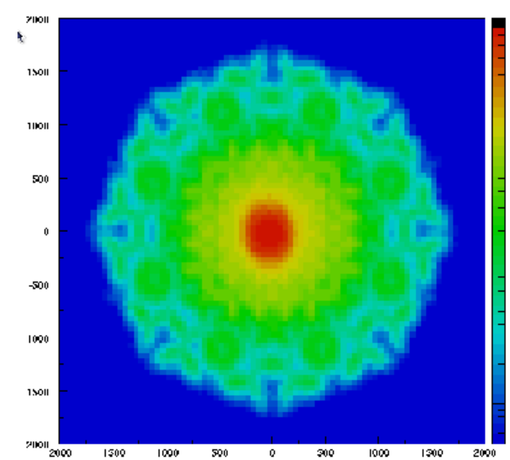
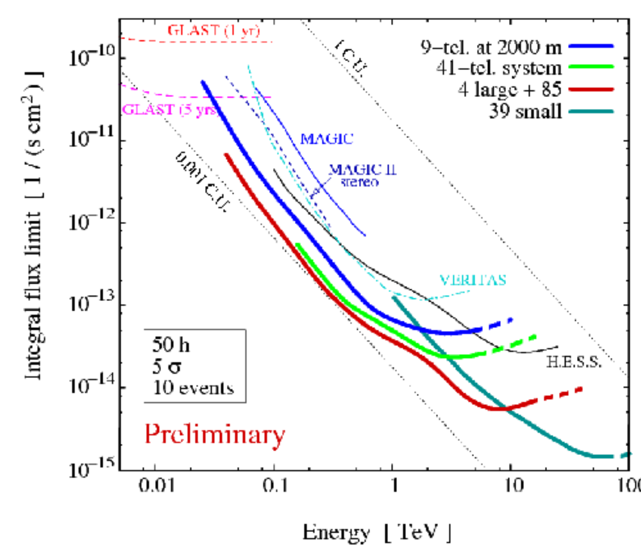
After the `CORSIKA / sim_telarray` Simulation Tool Chain: Analysis with `read_hess`

Konrad Bernlöhr

MPIK Heidelberg & HU Berlin







MC session at CTA meeting, Padua, 2008-11-03/05

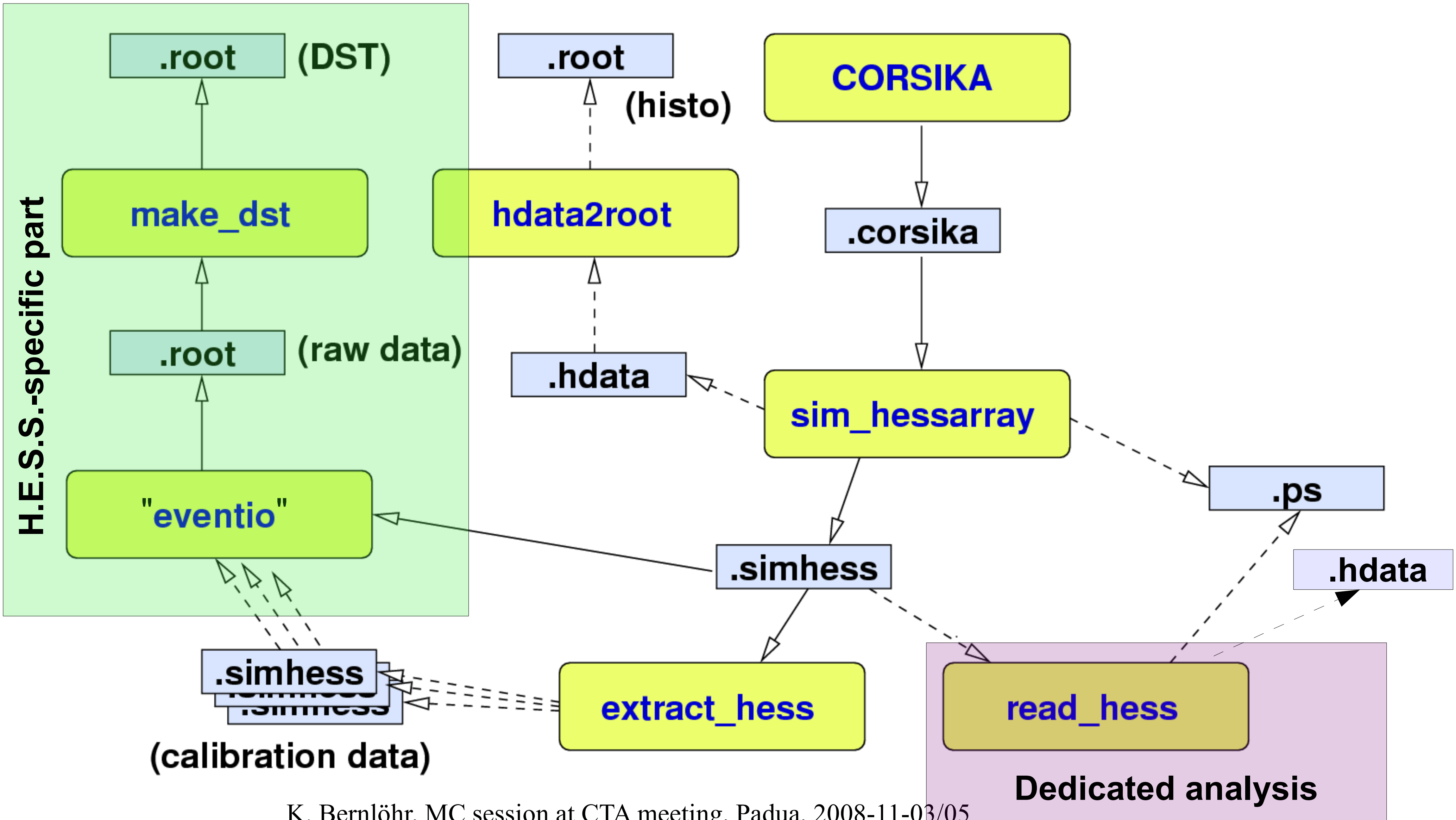
The CORSIKA / sim_telarray tool chain

- Shower simulation with **CORSIKA**.
  - Using the IACT option for machine-independent output based on the `eventio` library.
  - Telescopes are just defined there as fiducial spheres.
  - Output may be saved to disk or piped directly into one or multiple telescope simulations (which has basically no I/O bottlenecks).

The CORSIKA / sim_telarray tool chain

- **sim_telarray** output also eventio-based:
  - Including raw data integrated or in sample mode.
  - With second moments based analysis results from integrated reconstruction or external reconstruction program.
  - Histograms can be converted to ROOT (or PAW).
- Data conversion tools available (e.g. -> H.E.S.S.).
- Dedicated analysis program for quick development: read_hess (in the following: “Heidelberg Analysis”).

The CORSIKA / sim_hessarray processing pipeline



Analysis options

- Options with hessio format data:
  - read_hess (quick and, hopefully, not too dirty own development). Mostly C, no external dependencies.
  - conversion to HESS data format (“Sash”, ROOT-based) and analysis with HESS software.
  - conversion to MAGIC format by Abelardo's convertor.
  - write your own convertor, starting from read_hess_nr.

read_hess

```
Syntax: /home/konrad32c/hess/hessio/bin/read_hess [ options ] [ - |
   input_fname ... ]
Options:
   -p ps_filename  (Write a PostScript file with camera images.)
   -r level        (Use 10/5 tail-cut image cleaning and redo
   reconstruction.)
                   level >= 1: show parameters from sim_hessarray.
                   level >= 2: redo shower reconstruction
                   level >= 3: redo image cleaning (and shower
   reconstruction
                               with new image parameters)
                   level >= 4: redo amplitude summation
                   level >= 5: PostScript file includes original and
                               new shower reconstruction.
   -v              (More verbose output)
   -q              (Much more quiet output)
   -s              (Show data explained)
   -S              (Show data explained, including raw data)
   ...
```

And many more options. Options in red not available with the reduced version read_hess_nr. Both full and reduced version are now distributed within CTA.

read_hess example

```
opts=”-r 2 -u -q ${cuts} ....” # e.g.: --not-telescope 5,6,7,8
gamma_dst="../DST/gamma_${t1}_${t2}_-2.57.simhess-dst.gz"
proton_dst="../DST/proton_${t1}_${t2}_-2.70.simhess-dst.gz"
electron_dst="../DST/electron_${t1}_${t2}_-3.30.simhess-dst.gz"

read_hess ${opts} --powerlaw -2.57 --auto-lookup "${gamma_dst}"
read_hess ${opts} --powerlaw -2.57 --auto-lookup "${gamma_dst}"
read_hess ${opts} --powerlaw -2.57 "${gamma_dst}"

read_hess ${opts} --powerlaw -2.70 --theta-scale 6 "${proton_dst}"
read_hess ${opts} --powerlaw -3.30 --theta-scale 6 "${electron_dst}"
```

The read_hess output

Output from read_hess may include:

- Histograms (again in eventio/hessio format, can be converted to ROOT, HBOOK, eventio/iactio), some used as lookups for further analysis.
- Reconstructed/selected data (DSTs etc.)
- Text-mode n-tuples of MC/image/shower data.
- Postscript camera images.
- All the data explained in full detail.

Heidelberg CTA-MC Analysis 1

- Simple analysis (with updates since last meeting):
  - Cleaning: 4/7, 5/10, 8/12 p.e. tail-cuts and other scheme;
  - Image amplitude > 30 p.e., up to 200 p.e.
  - Image c.o.g. Radius + 0.5 image length < 0.85 R_{cam} and variations of it ("edge cut"). Potential problem at high *E*.
  - Geometrical shower reconstruction (direction and core position) from Hillas parameters, using weighted mean of pair-wise intersections of image major axes.
    This is certainly sub-optimal and can be improved!

Heidelberg CTA-MC Analysis 2

- Generating lookup tables width+length of gamma rays.
- Using the lookups, get mean reduced scaled width+length

$$mrscw = \frac{1}{N} \sum_{i=1,N} \frac{(w_i - \bar{w}(r_c, A, z))}{\sigma_w}$$

  and cut on *mrscw* and *mrscl* for gammas and protons ("shape cuts"). Energy dependent cuts.
- Get angular resolution and apply point source selection ("angle cut"). Multiplicity-dependent cut.

Heidelberg CTA-MC Analysis 3

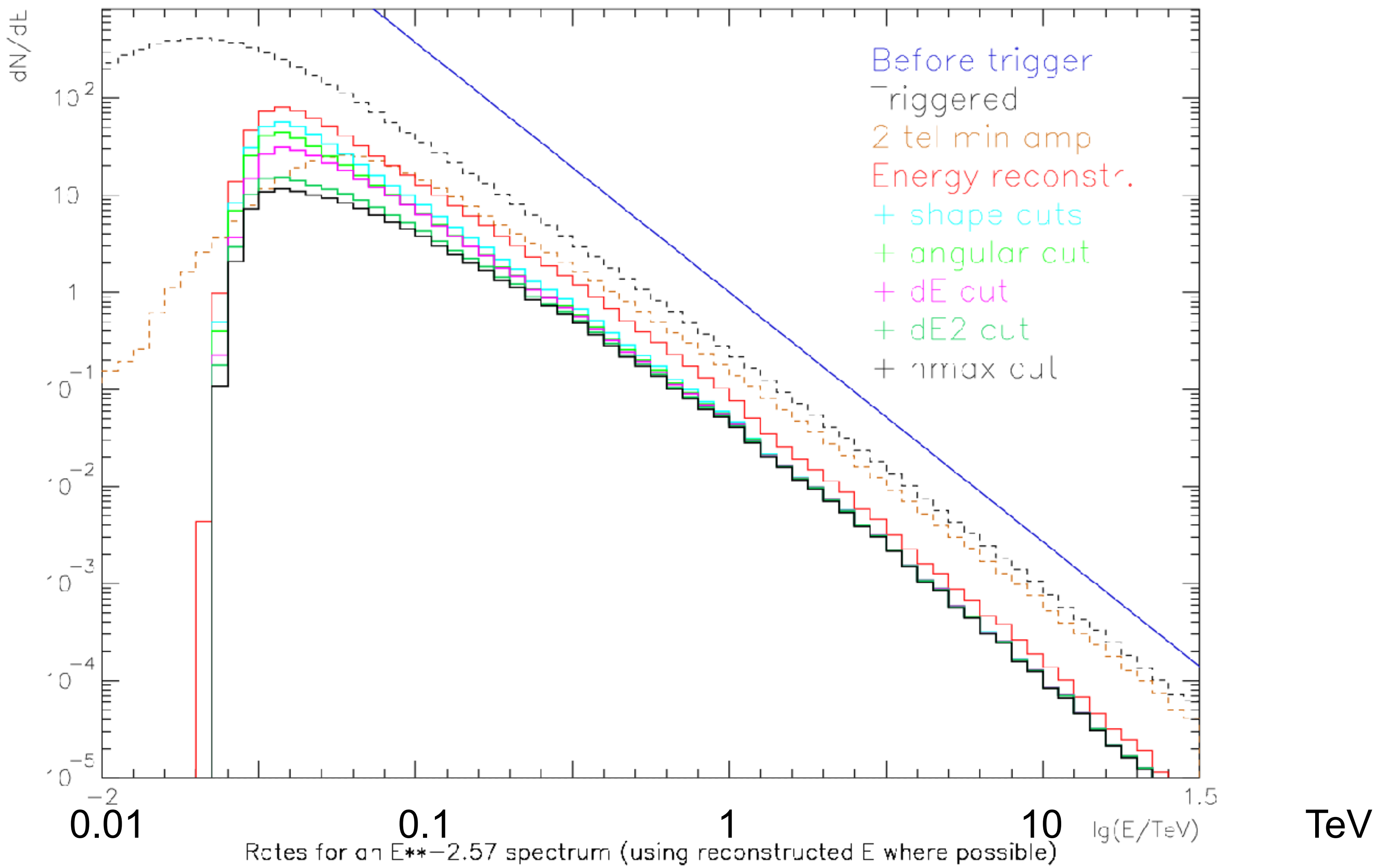
- Generating lookup tables for image amplitude / energy (*I/E*) as with width and length.
- Using that lookup an energy estimate and estimate of its fluctuation for each telescope is obtained.
- Get energy E, energy accuracy σ_E/E, and χ^2/n.d.f.
- Discard showers with bad σ_E/E ("dE cut").
  Rejects also gammas at large (but energy dependent) R_c.
- Discard showers with bad χ^2/n.d.f. ("dE2 cut").
- Calculate distance of shower maximum and discard showers inconsistent with gammas ("hmax cut").

Heidelberg CTA-MC analysis 4

New in the eventio/hessio format:

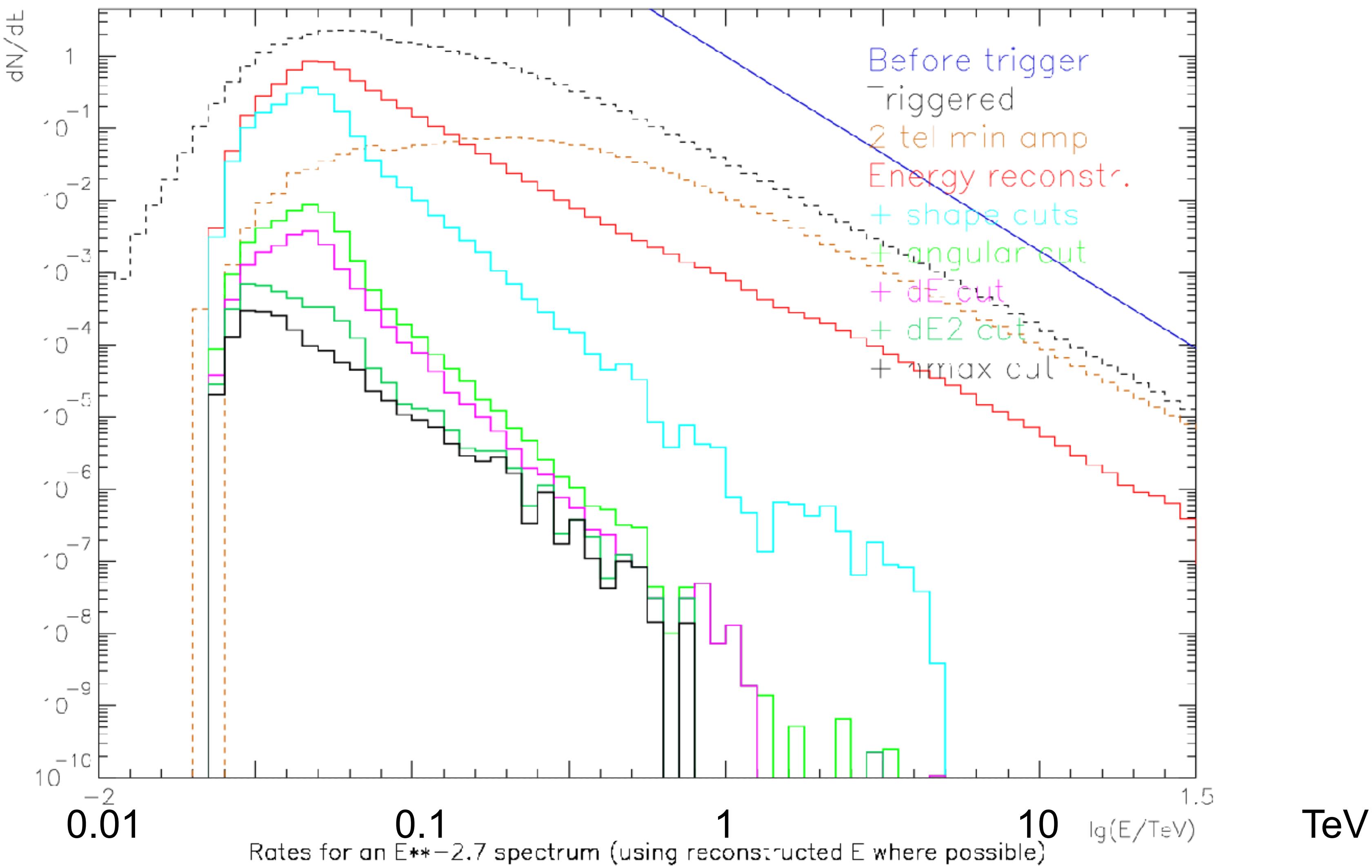
- Inter-telescope trigger time difference w.r.t. nominal shower plane.
- “Online” pixel pulse shape analysis (peak, rise, width). For pixel amplitudes above threshold.
- Pulse shape analysis summary at DST level: peak time gradient (along major axis), r.m.s. residuals, mean pixel pulse widths (20%, 50%), mean rise time.

Cut efficiencies (gammas)



K. Bernlöhr, MC session at CTA meeting, Padua, 2008-11-03/05

Cut efficiencies (protons)

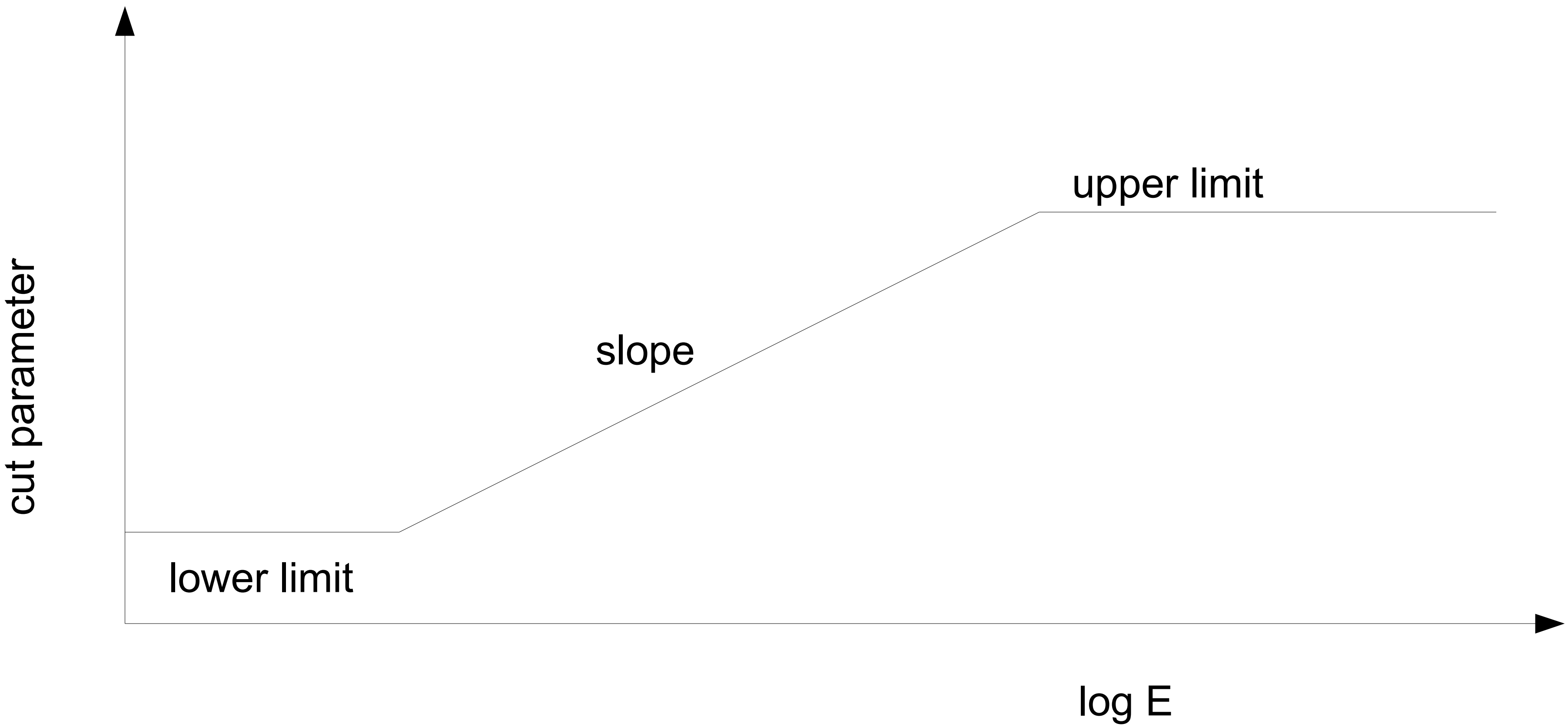


K. Bernlöhr, MC session at CTA meeting, Padua, 2008-11-03/05

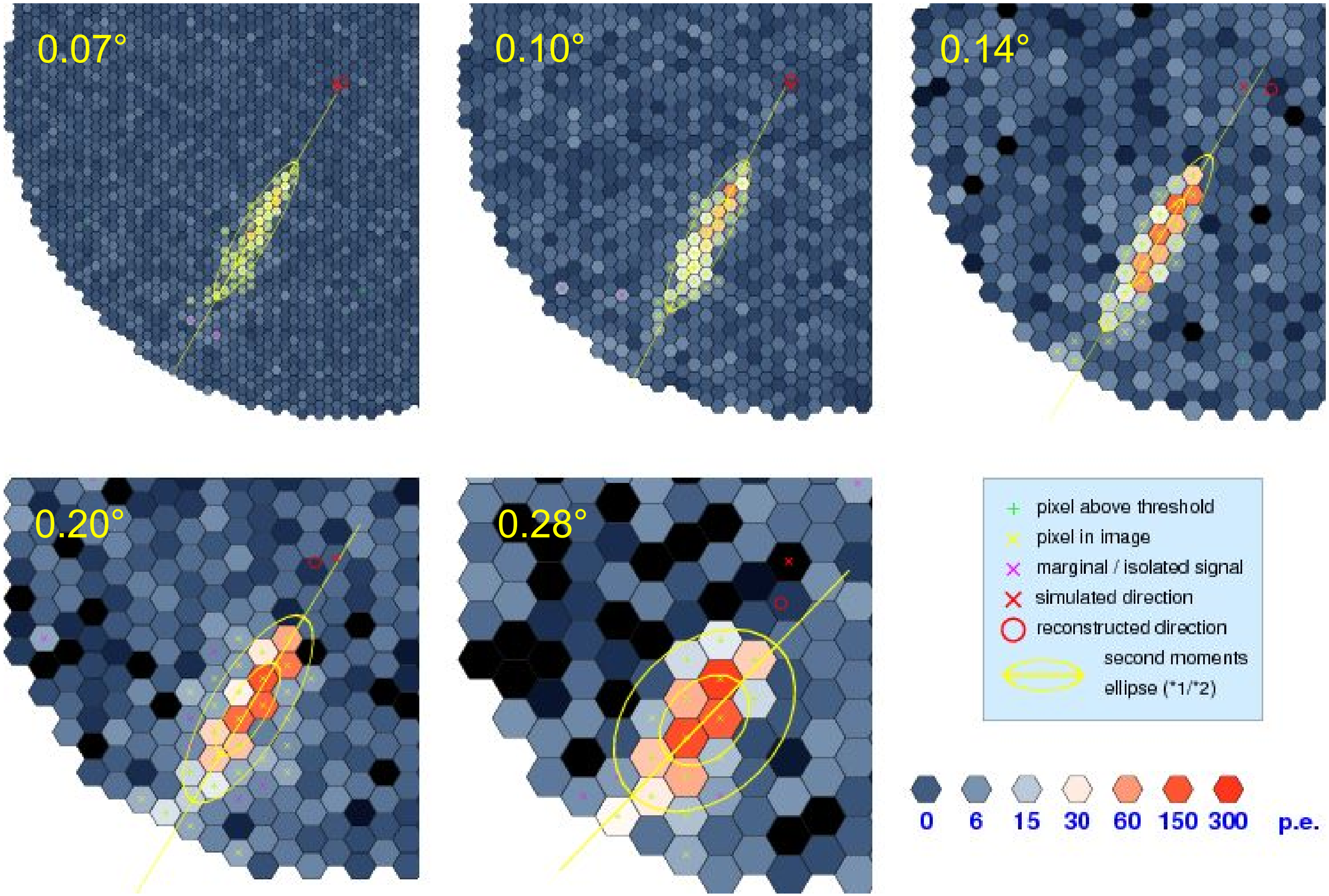
Optimizing shape cuts etc.

- Background falls off rapidly at higher energies.
- Optimum shape cuts are, compared to fixed cuts:
  - more strict at low energies (better hadron rejection)
  - less strict at high energies (more gamma signal)
- Similar for additional cuts like
  - *dE* cut (accepting events with well-determined energy),
  - *dE2* cut (events with consistent energy estimates from individual telescopes), and
  - *hmax* cuts (height of maximum as expected for gammas) .

Allowed dependence of cut parameters

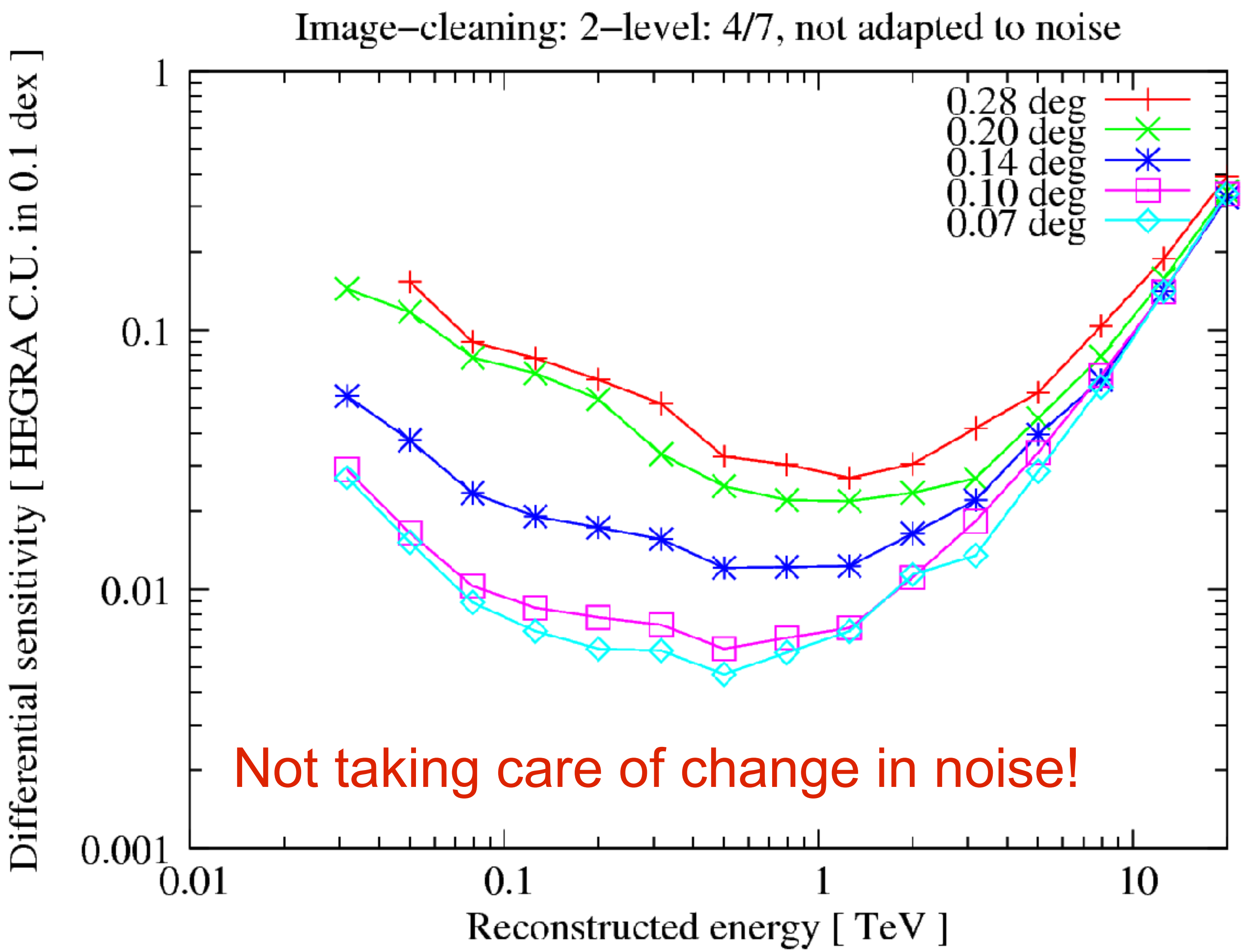


Example: Impact of pixel size



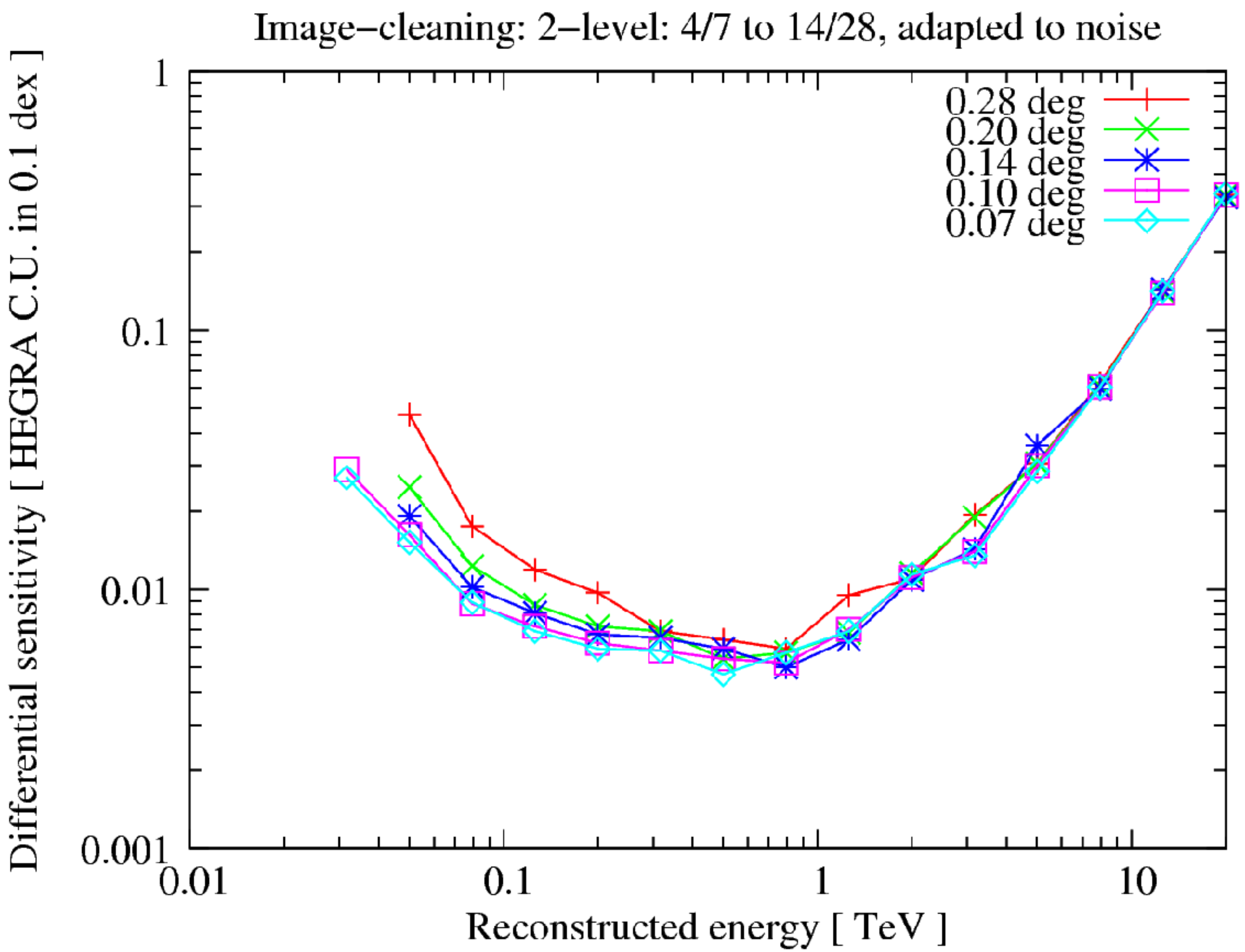
K. Bernlöhr, MC session at CTA meeting, Padua, 2008-11-03/05

Pixel size: Sensitivity



K. Bernlöhr, MC session at CTA meeting, Padua, 2008-11-03/05

Pixel size: Sensitivity

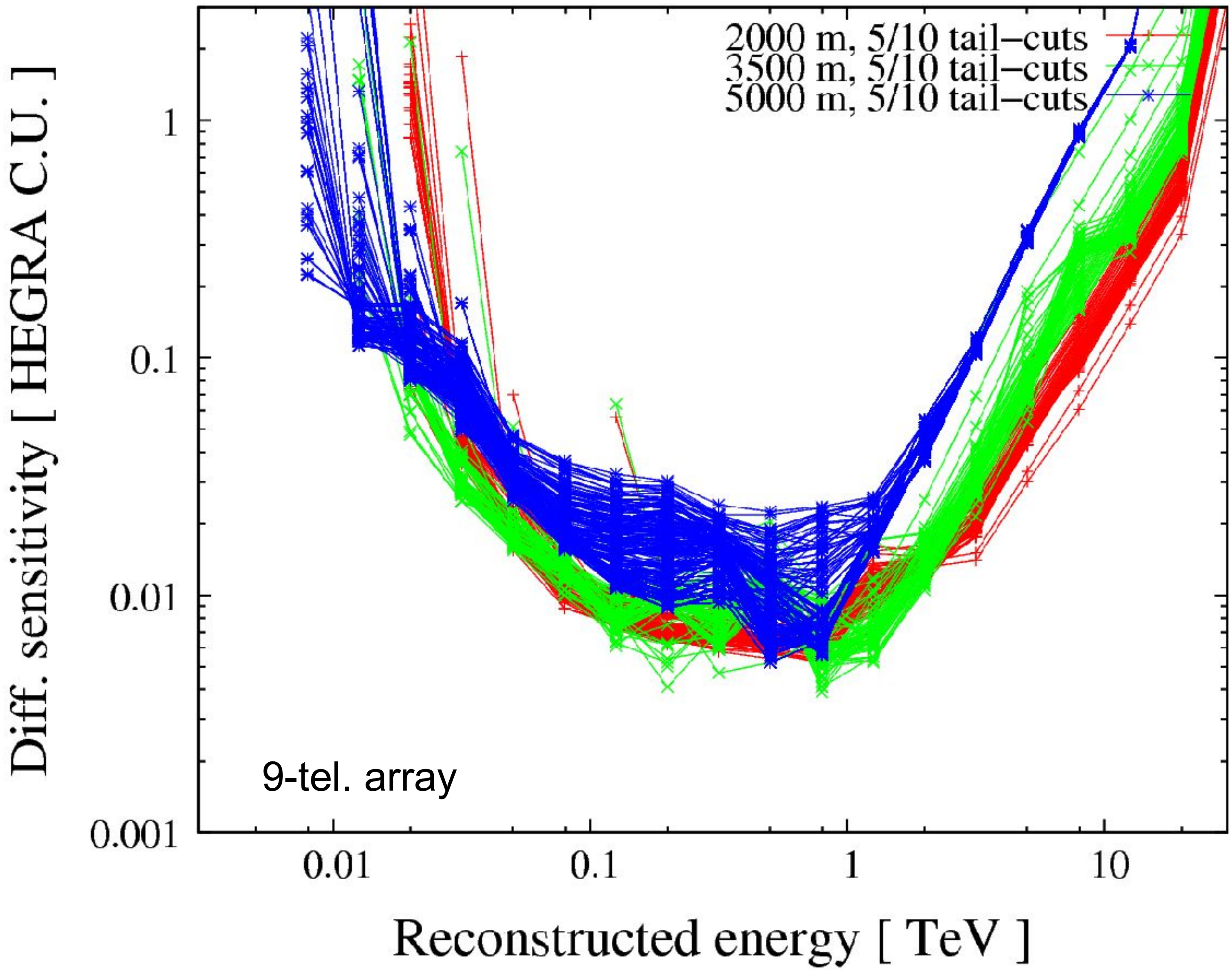


K. Bernlöhr, MC session at CTA meeting, Padua, 2008-11-03/05

Optimum pixel size ?

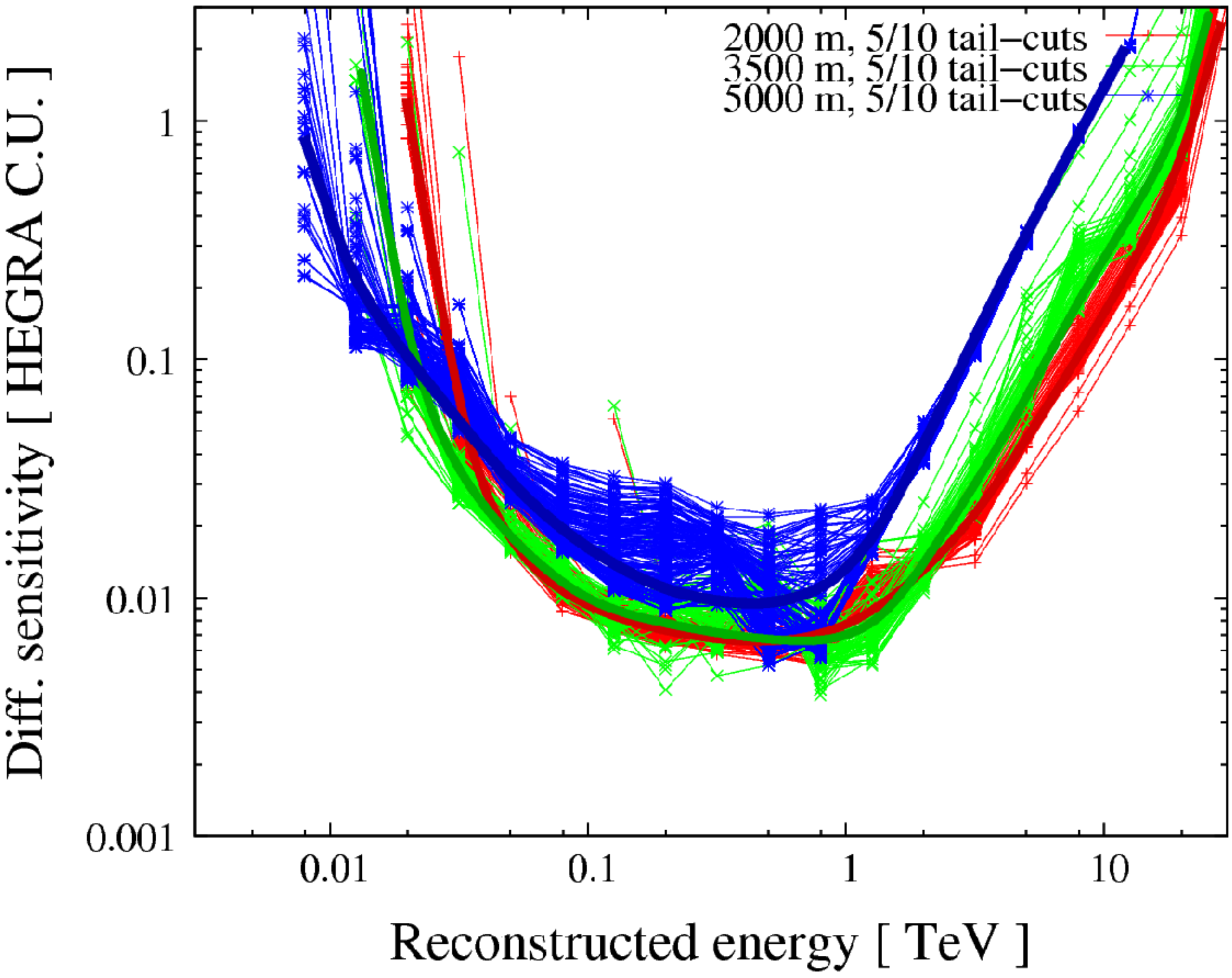
- Smaller pixels resulting in slightly better angular resolution – but image cleaning must really be optimized separately for each pixel size.
- At high energies, no improvement seen with smaller pixels.
- Only a modest improvement at low energies.
- The initial assumption of 0.10° pixels for the large telescopes was very reasonable.
- Cost optimization may favour larger pixels.

Example: 2000 m + 3500 m + 5000 m



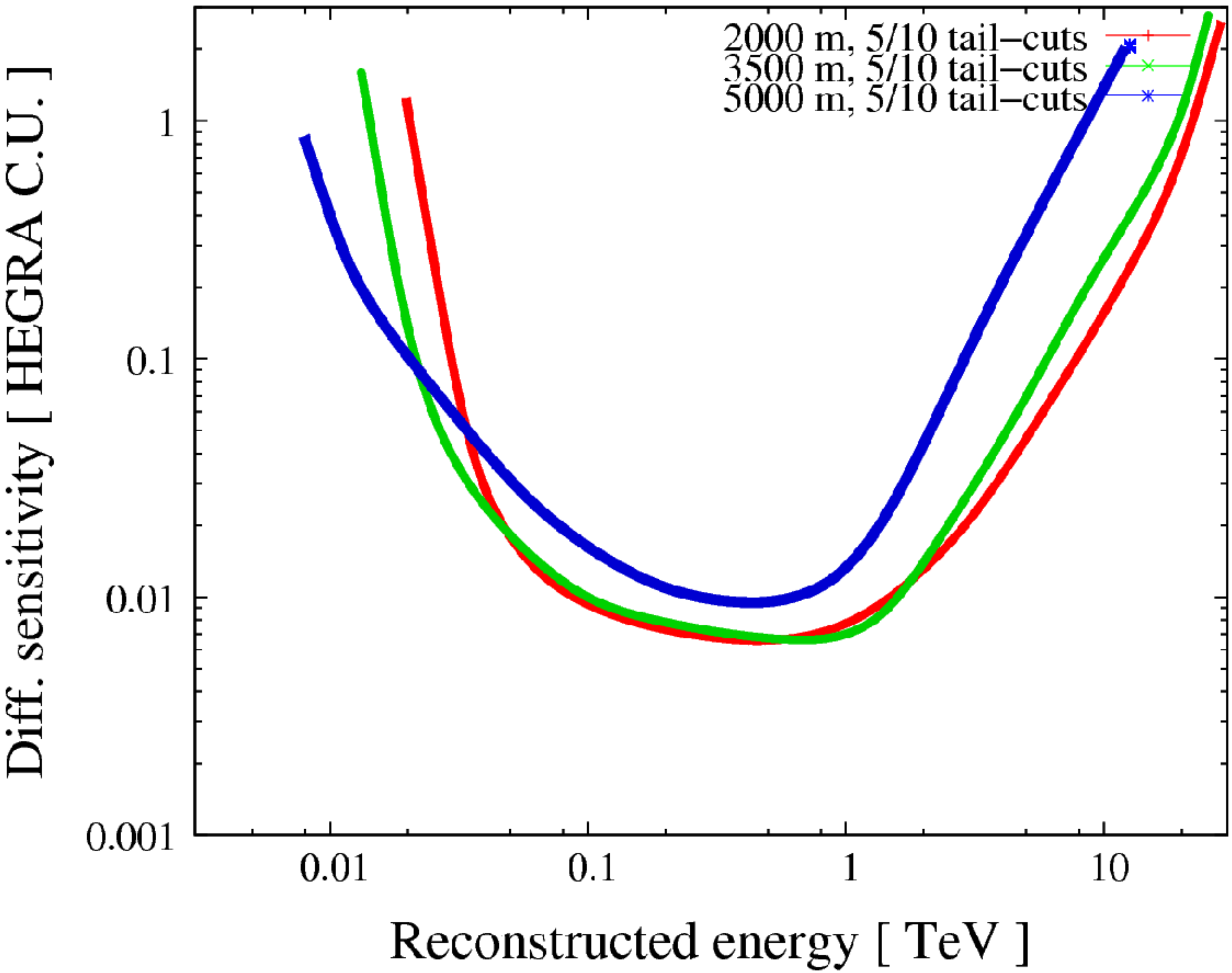
K. Bernlöhr, MC session at CTA meeting, Padua, 2008-11-03/05

2000 m + 3500 m + 5000 m



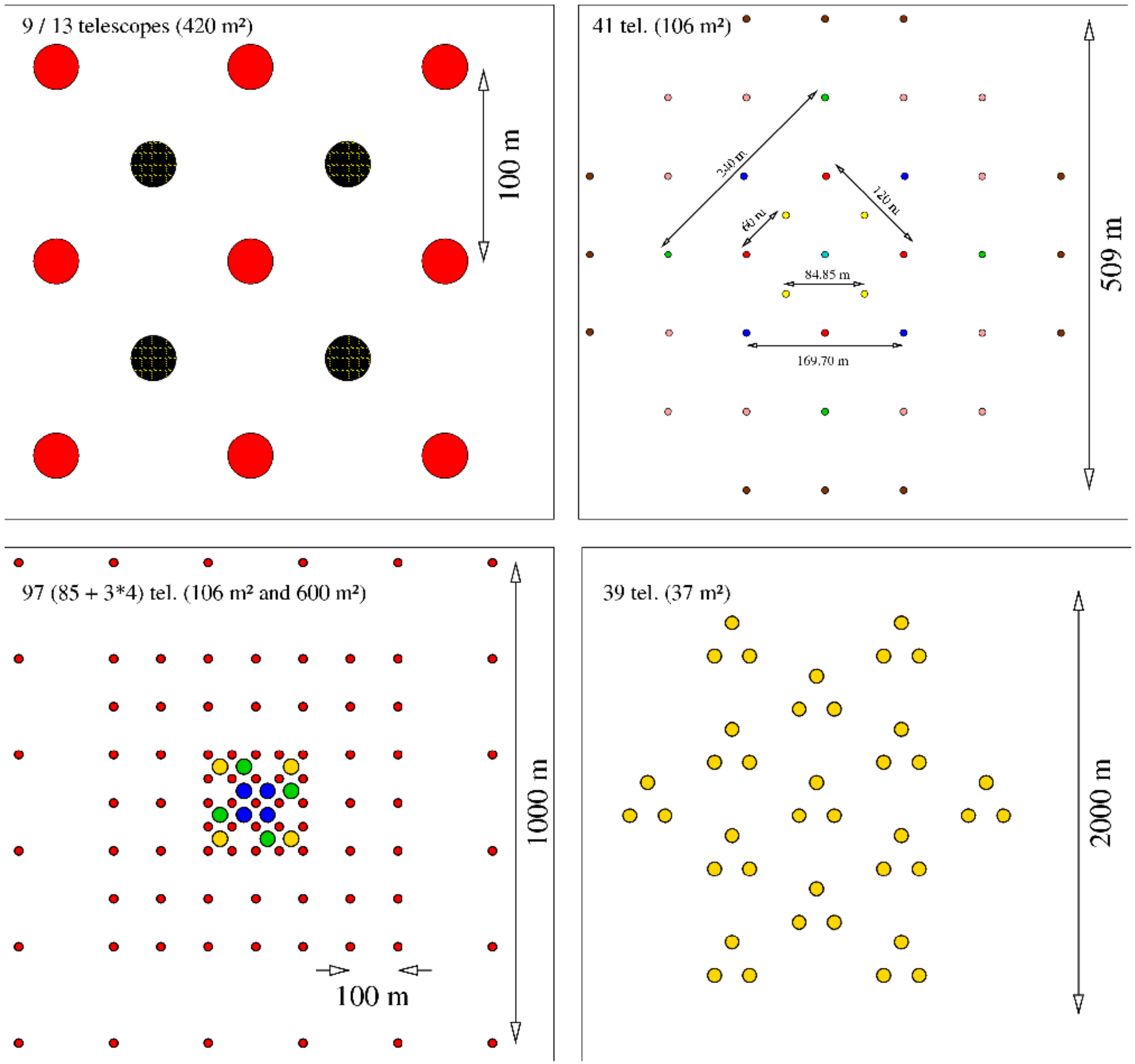
K. Bernlöhr, MC session at CTA meeting, Padua, 2008-11-03/05

2000 m + 3500 m + 5000 m

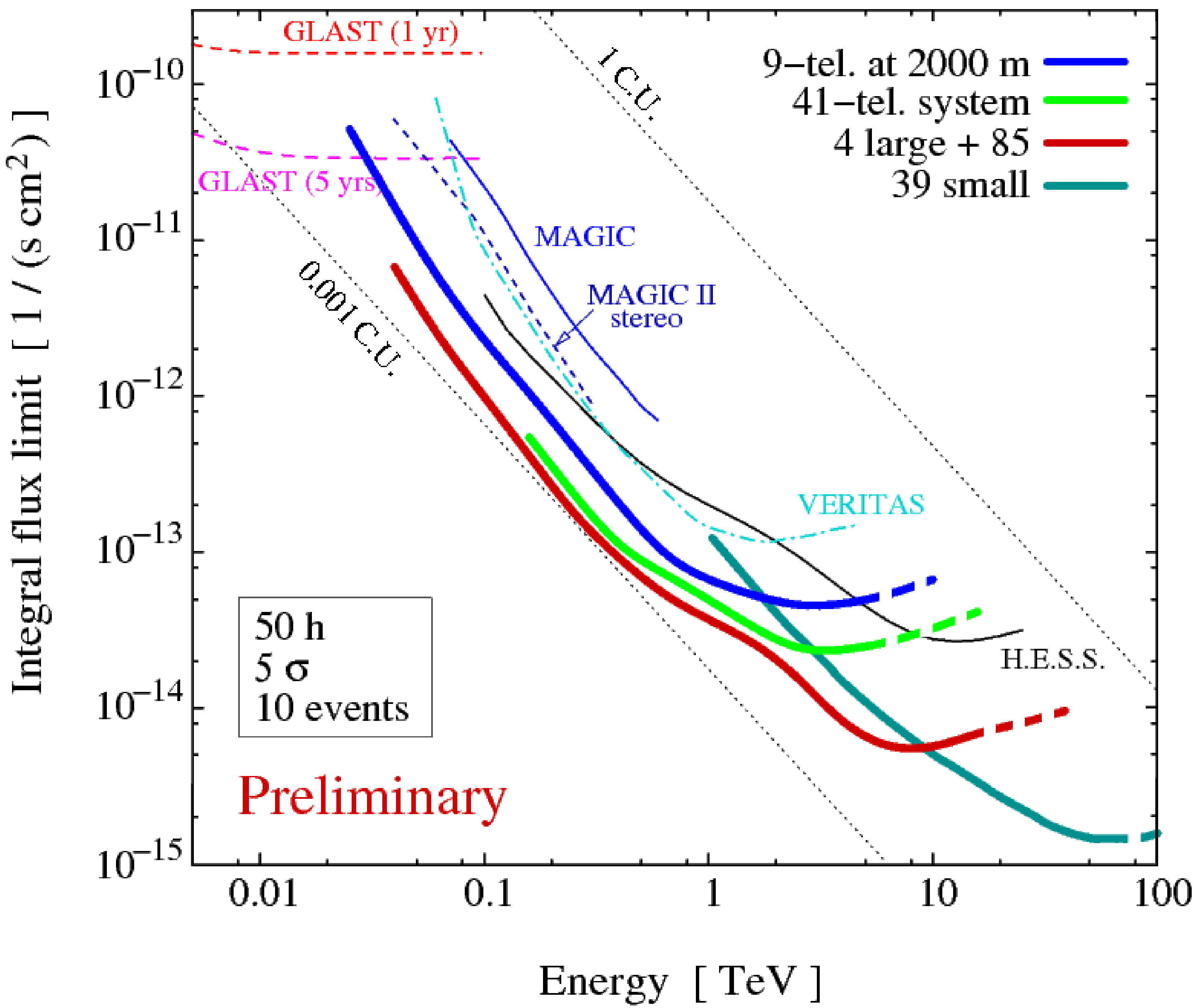


K. Bernlöhr, MC session at CTA meeting, Padua, 2008-11-03/05

Example: A variety of arrays



Example: A variety of arrays



Additions needed for “Ultra” data sets

- Lookups used to be grouped by mirror area. Now we may have different telescope types with the same mirror area.
- Different tail-cuts are needed for the different telescope types. “Auto”-adapting tail cuts?
- Taking more care of saturated images.
- Cut on shower direction (“Θ^2 cut”) no longer just depending on multiplicity, ...

Conclusions

- Analysis in the read_hess framework does not depend on any external packages.
- Quick development and quickly adapted.
- Mostly like H.E.S.S. Hillas-based standard analysis, with additional bells and whistles.
- Simple (partly manual) cut optimization.
- Analysis and cuts may leave room for improvements.
- **You** should demonstrate that you can beat its performance in terms of sensitivity / angular resolution / energy resolution / hadron rejection, ...