An introduction to

MEGAlib

The Medium-Energy Gamma-ray Astronomy library

Andreas Zoglauer

University of California at Berkeley

Space Sciences Laboratory

Intro

What is MEGAlib?

- A C++ library with python3-bindings and a set of programs for gamma-ray instrumentation.

What is it good for?

- Cover the whole path from simulations/measurements, via calibration, event classification, to high-level data analysis for astrophysics (primary goal) and for terrestrial applications (environmental monitoring, nuclear medicine, etc.) in hard X-ray to medium-energy gamma-ray region.
- Most high-level analysis tools concentrate on Compton telescopes

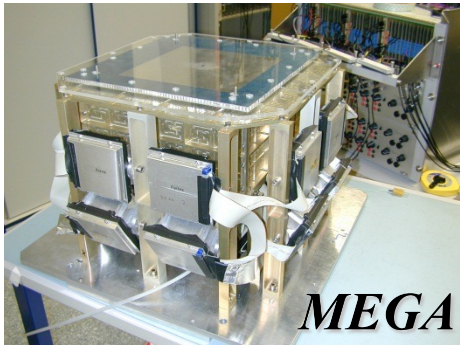
What are MEGAlib's design goals?

- Applicability to all measurement scenarios for hard X-ray and gamma-ray space instruments
- Try to be as easy to use as possible
- Split UI and library to be highly versatile and reusable

Past and current development drivers of MEGAlib

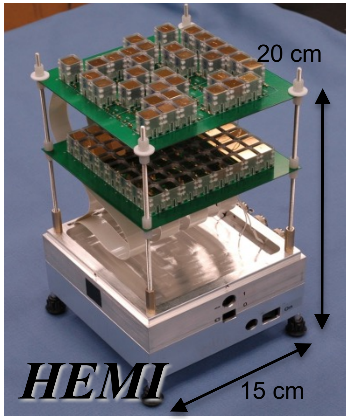
Starting in 1999, MEGAlib was developed as simulation and analysis tool for the combined Compton scattering and pair creation telescope MEGA at MPE/Garching.

However, its flexible design allowed its application to many more detector types and instruments.



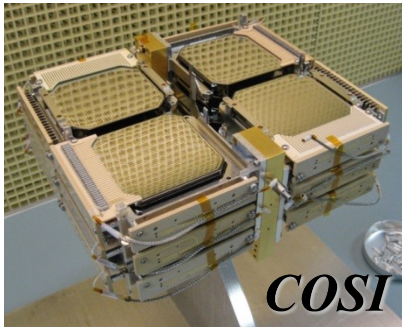
The main development drivers were/are:

- **MEGA / GRIPS / AMEGO:** Combined Compton-scattering and pair-creation telescopes (1999 – today)
- **HEMI**: A combined coded-mask and Compton camera (2008-2014)
- **COSI:** A cross-strip germanium-based Compton telescope (2004 – today)

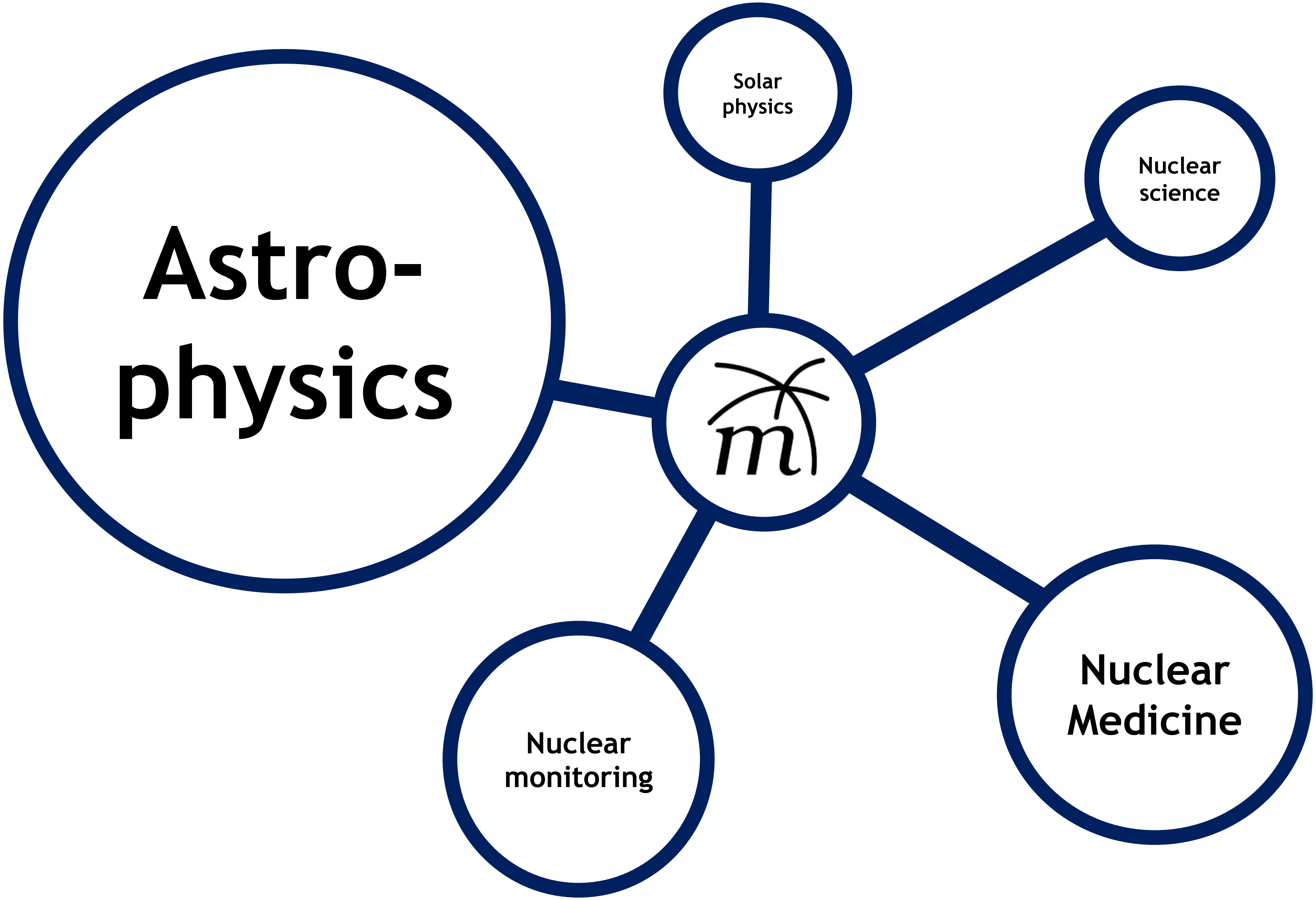


MEGAlib has been applied to many instruments:

- ACT, NuSTAR, COMPTEL, AMEGO(-X)/ComPair, (e)ASTROGAM, GRI/DUAL, GRIPS@SSL, TIGRE, TGRS, XFEL & hadron therapy detector development simulations, several homeland security projects, and many others



MEGAlib across Domains

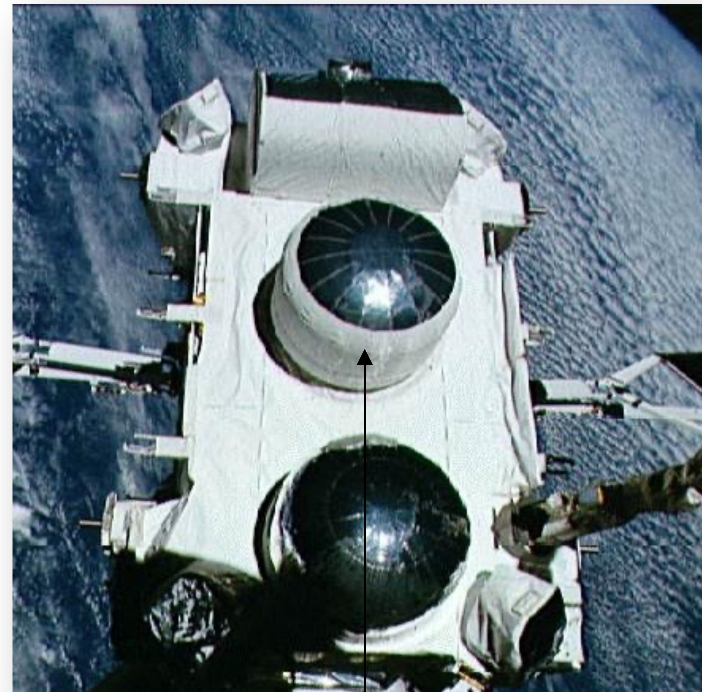


Application areas: Astrophysics

COSI/NCT balloon flights (2004, 2009, 2014, 2016) and space mission (2027+)

COSI

Data from the COMPTEL archive (1991-2000)



CGRO/COMPTEL

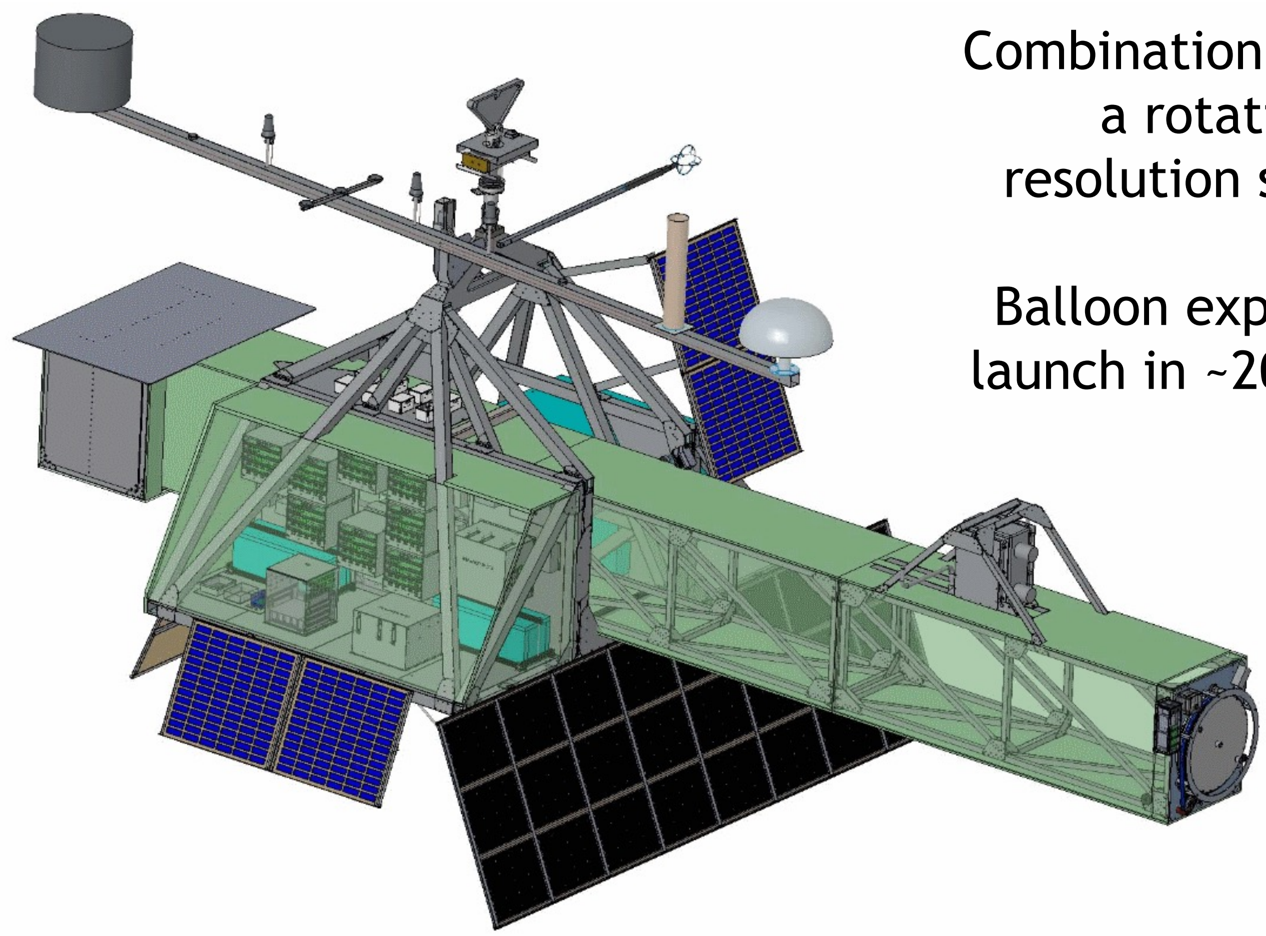
COSI's upcoming space mission is the current main driver for MEGAlib's development, besides an APRA to develop machine learning tools

Application areas: Solar Physics

GRIPS - Gamma-Ray Imager/Polarimeter for Solar flares

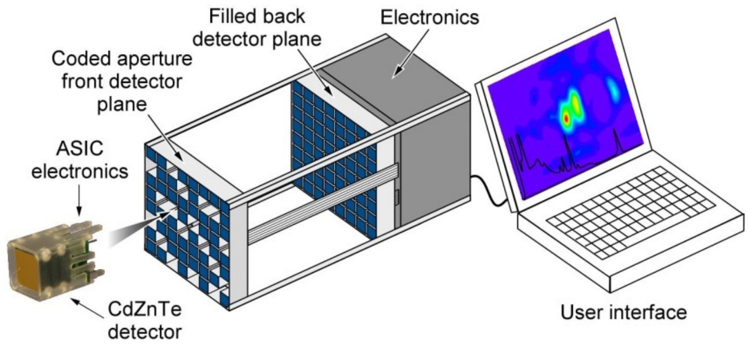
Combination of COSI detectors with a rotating modulator for high-resolution solar flare observations

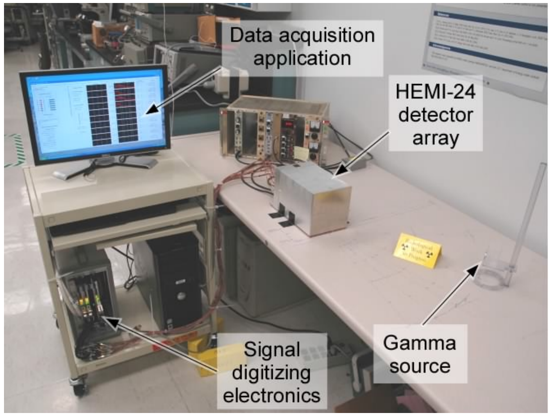
Balloon experiment with expected launch in ~2024/25 from Antarctica



Application areas: "Environmental monitoring"

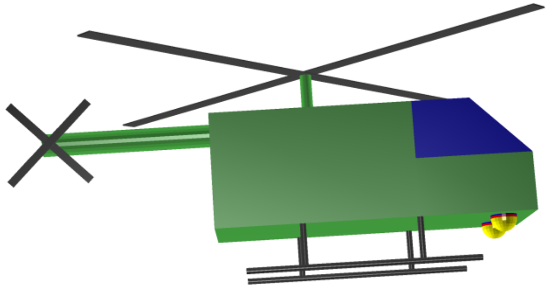
HEMI – the High Efficiency Multimode Imager



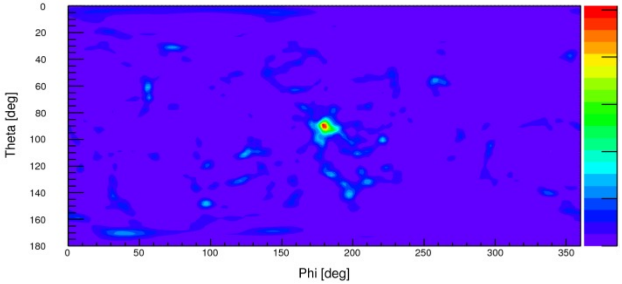


Consists of multiple 1-cm³ CPG-CZT detectors (2% FWHM @ 662 keV)

Simulation & imaging application example:



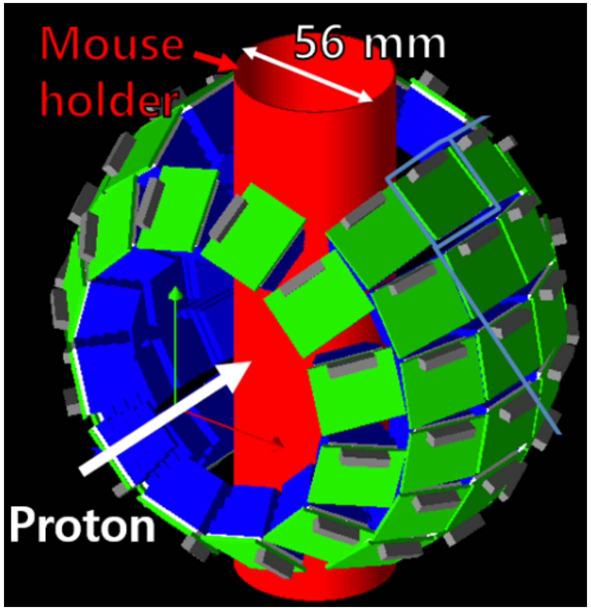
2 half-spherical HEMI instruments below helicopter (1500-CZT-detectors each)

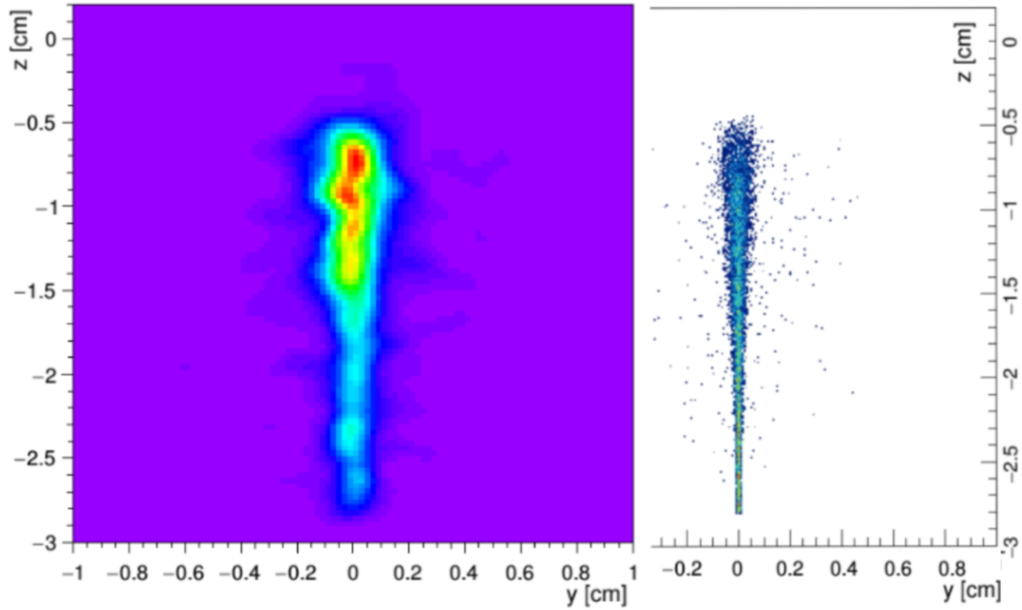


Reconstructed image of a 1 mCi ^{137}Cs source 50 m below the helicopter after 20 sec.

Application areas: Medical imaging

Hadron therapy monitoring





PET (Positron-emission tomography) detector system for mouse imaging modelled and simulated in MEGAlib

Left: 3D PET image reconstruction with MEGAlib
Right: origins of positrons in proton beam

From: Giulio Lovatti et al. 2020

Technical

Webpage – Or just google „MEGAlib"

http://megalibtoolkit.com

https://github.com/zoglauer/megalib

License

Open source - version 3: GPL v3, version 4: LGPL v3

Latest versions in git

Official release: 3.06 – Development version: 3.99.01 (main branch)

Programming language

C++ and based on ROOT and Geant4 + python3 bindings

Programming philosophy

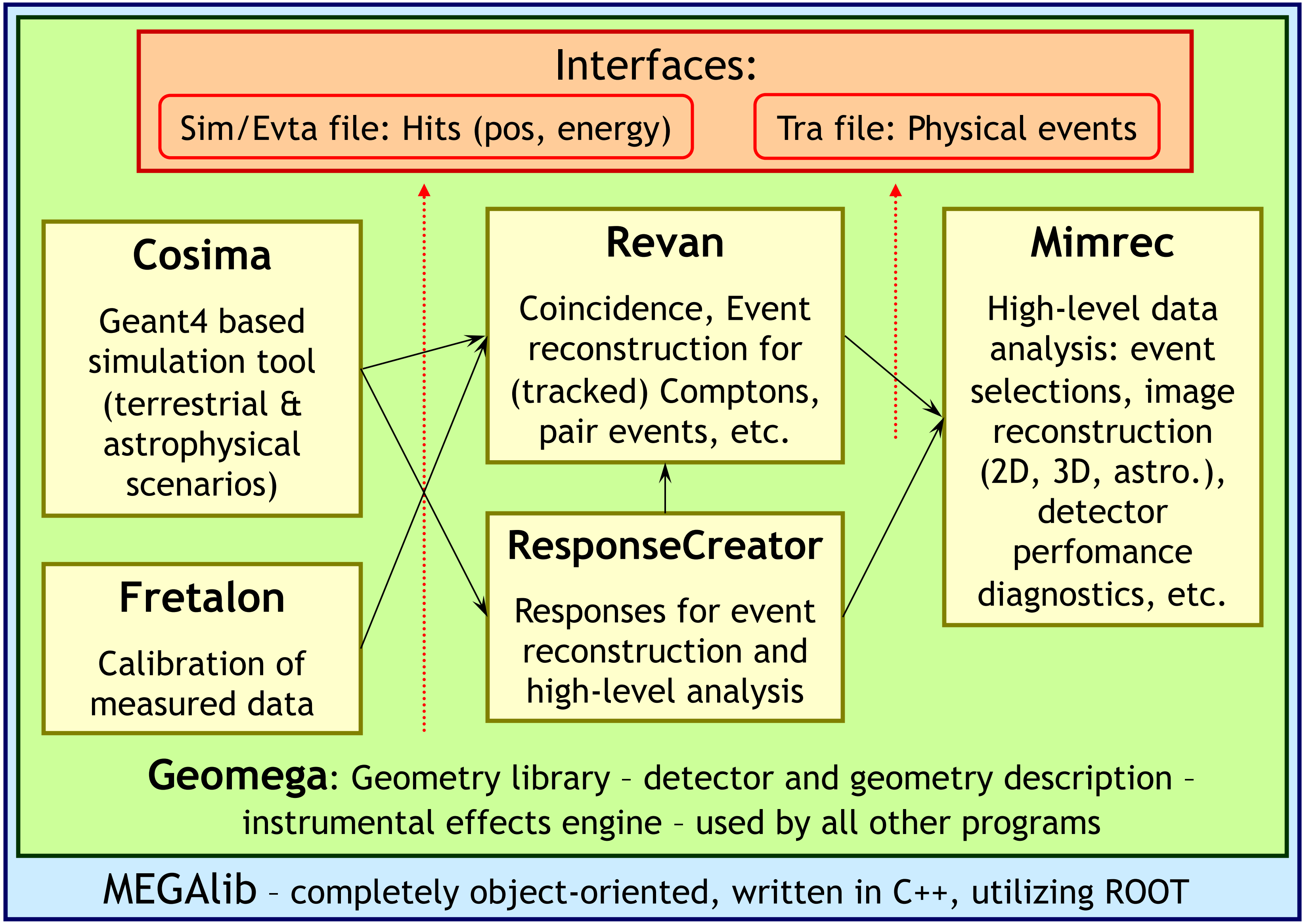
Object-oriented library with GUI programs utilizing the library

Easy to write additional analysis programs in C++ or Python utilizing MEGAlib

For bugs, comments, suggestions, volunteered code, etc. use the GitHub issue tracker or email

Andreas Zoglauer (zoglauer@berkeley.edu)

MEGAlib – Main Tools



The Standard Workflow: Simulation

Build geometry & define detector properties

Materials, volumes, placements, detectors, trigger criteria

↓

Simulation & Detector Effects Engine

Define environment & perform Geant4 simulation & digitize and noise

↓

Event classification

Identify Compton, pair sequence and overall event quality

↓

Data Analysis

Imaging, spectral identification, polarization, sensitivity determination, etc.

The Standard Workflow: Measurements

Build geometry & define detector properties

Materials, volumes, placements, detectors, trigger criteria

Perform measurements with your detector

Record strips/voxels, AD units, timings, times, etc.

↓ ↓

Event Calibration

Utilize Fretalon framework for energy, position, etc. calibration

↓

Event classification

Identify Compton, pair sequence and overall event quality

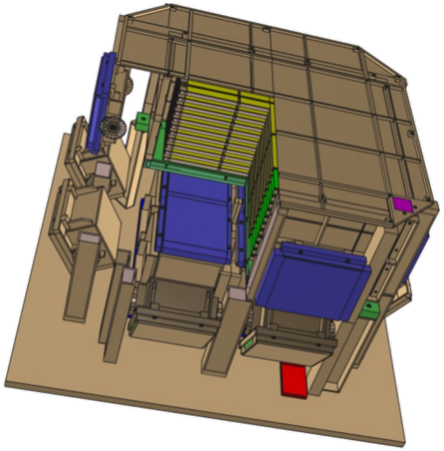
↓

Data Analysis

Imaging, spectral identification, polarization, sensitivity determination, etc.

Core Tools: Geomega

- Provides geometry and detector description for MEGAlib
- Additional features:
  - Detector effects engine
  - Display of geometry
  - Conversion to MGGPOD & Geant4 format
  - Absorption probabilities
- Described by setup file
  - Keyword base
  - Object oriented
  - Constants, maths environment, for-loops, if-conditions, etc.



Cut-away view of MEGA detector using Geomega & ROOT

```
// One Si-Wafer of the tracker
Volume Wafer
Wafer.Material Silicon
Wafer.Visibility 1
Wafer.Color 2
Wafer.Shape BRIK 3.15 3.15 0.025

Wafer.Copy WaferN01
WaferN01.Position  8.45 8.45 0.0
WaferN01.Mother SiStripDetector


Wafer.Copy WaferN02
WaferN02.Position  8.45 2.05 0.0
WaferN02.Mother SiStripDetector
```

Why do we need a geometry?

A geometry provides information on:

- Where is which volume?
- Which material does it consist of?
- Which volumes are active detectors and what are its properties?
- What are the trigger criteria?

Required for: Everything

- Calibration
- Simulation
- All data analysis tasks

Key elements

Key elements of the geometry file:

- Material definitions
- Volume definitions (boxes, spheres, tubes, cones, trapezoids, polygone, polycones, etc. plus boolean operations on those)
- Volume placements
- Detector definitions (Simple one detectors, 2D/3D voxel, 2D/3D strip, Anger camera, COSI/AMEGO-like calorimeter) including resolutions & thresholds
- Trigger & Veto definitions

Plus:

- Constants
- Maths operations
- if-clauses, for loops, etc.

Geomega's Detector Effects Engine

Makes sure the simulations look close to the measurements.

Handles:

- Energy resolution (Gauss, Lorenz, Gauss-Landau)
- Depth resolution
- Time resolution
- Noise thresholds
- Trigger threshold
- Overflow (considered AD converter overflow)
- Energy loss map: 3D voxel map of energy scalers (e.g. 0.8 → only 80% of the energy in the voxel are measured, rest is e.g. trapped)

Geomega's detector effect engine is just an initial approximation. Your final detector effects engine must be written by yourself and exactly invert your calibration detector by detector, pixel by pixel, and the data must then be run through your calibration pipeline

Core tools: Cosima - Simulation

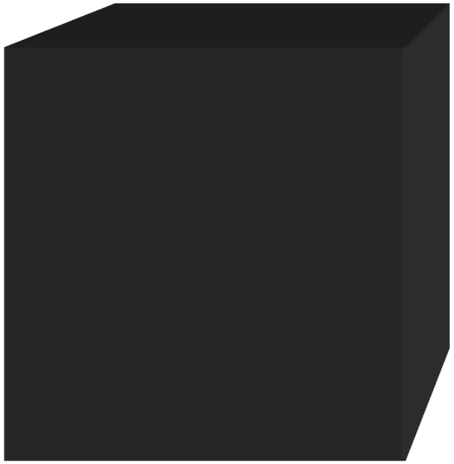
Create a simulation input file
(which particles from where, when, with which energy and polarization)

Launch cosima:

```
user:/> cosima –z MyInputFile.source
```

Particles are simulated…

The Monte-Carlo simulations are a big, black box for the end user…



User does not need to know any details about Geant4 simulation

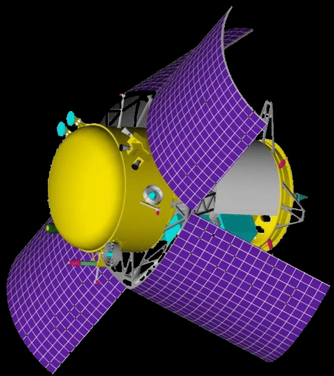
Results are stored in a simple ASCII output format compatible with all other MEGAlib tools (can be automatically created gzip'ed)

Main application: The Space Environment

Sun through solar flares: photons, charged particles

Radiation belts:

Trapped protons (SAA) & resulting activation, electrons



Cosmic rays:

- Protons (& activation)
- Alphas
- Ions
- Electrons
- Positrons

For some applications cosmic photons are also background

Secondaries induced by cosmic ray interactions with upper atmosphere:

Leakage photons, neutrons, protons, electron, positrons, etc.

Cosima input format

```
Version             1

Geometry            Satellite.geo.setup

PhysicsListEM       LivermorePol
PhysicsListHD       QGSP-BIC-HP


StoreSimulationInfo     init-only


Run SpaceSim
SpaceSim.FileName     CrabObservation
SpaceSim.Time         10000.0
```

Load Geomega geometry & detector info

Define physics lists

Define output options

Define a simulation run with file output name and stop criterion (time, started events or triggered events)

Cosima input format

```
SpaceSim.Source Crab
Crab.ParticleType 1
Crab.Beam FarFieldPointSource 0 0
Crab.Spectrum PowerLaw 100 100000 2.17
Crab.Flux 0.004972
Crab.Polarization 1.0 0.0 1.0 0.0 1.0
Crab.LightCurve File true Lightcurve.dat

SpaceSim.Source CosmicPhoton
CosmicPhoton.ParticleType 1
CosmicPhoton.Beam FarFieldAreaSource 0.0 113.4 0.0 360.0
CosmicPhoton.Spectrum File CosmicPhotonSpectrum.dat
CosmicPhoton.Flux 0.035984
```

Define multiple sources, each with:

- Particle type
- Beam option
- Spectral options
- Total flux
- Polarization
- Light curve

Beam & spectral types

Far field beam :

- Point source
- Gaussian
- Area source
- Zenith depended variation
- Combined 3D spectral-beam-flux file

Can be in instrument or Galactic coordinates

Near field beam:

- Point
- Line
- Box
- Sphere
- Disk
- Pencil beam (Homogenous, radial profile, profile defined by file)
- Cone Beam (flat, Gaussian)
- Illuminated disk/box
- Emission pattern from file
- Geometry volume (e.g. in combination with radioactive isotopes)

Spectra:

- Mono
- Linear
- Power law
- Broken power law
- Gaussian
- Blackbody
- File (lin-lin, lin-log, log-lin, log-log interpolation modes)
- Individual isotopes for detector activation simulations

Alternatively:

Event list generated by external tool

The *.sim simulation output file format

```
SE
ID    4  568
TI    5.535837965
IA INIT  1; 0;0;0.000000000000e+00;  12.35804;  11.28213;  40.00000;0;
   0.00000; 0.00000; 0.00000; 0.00000; 0.00000; 0.00000;     0.000;1;-
   0.00000;-0.00000;-1.00000; 0.00000; 0.00000; 0.00000;   805.798
IA COMP  2; 1;1;2.034125846877e-09;  12.35804;  11.28213; -20.98156;1;-
   0.04567; 0                                              .722;3;
   0.39452;-0                                              .977
IA PHOT  3; 1                                              1599;1;
   0.00000; 0                                              .000;3;-
   0.58865; 0                                              .558
IA PHOT  4; 1                                              .51599;1;
   0.00000; 0.00000; 0.00000; 0.00000; 0.00000; 0.00000;     0.000;1;
   0.26593; 0.96265;-0.05082; 0.00000; 0.00000; 0.00000;    28.610
<…cut…>
HTsim 1;  12.34950;  11.26850; -21.00000;   8.07601;0.00000e+00;2
HTsim 2;  11.32000;  13.48000; -40.53665; 628.02789;0.00000e+00;5;6;7;8;9
HTsim 2;  10.78000;                                  00e+00;10
HTsim 2;  10.24000;                                  00e+00;11;12;13
```

Event ID

Observation time

Ideal simulation information:
Path of the original photon and secondary photons
→ Used for diagnostics and response generation

Measurement-like data before noising:
→ Positions and energy deposits

Using the simulation data

The moment the simulation file is read in within MEGAlib, Geomega's detector effects engine is applied to:

- Noise energies, depth & times
- Apply thresholds and trigger criteria

At this stage the simulation data should resemble the measurement data, and MEGAlib's high level analysis parts can be applied:

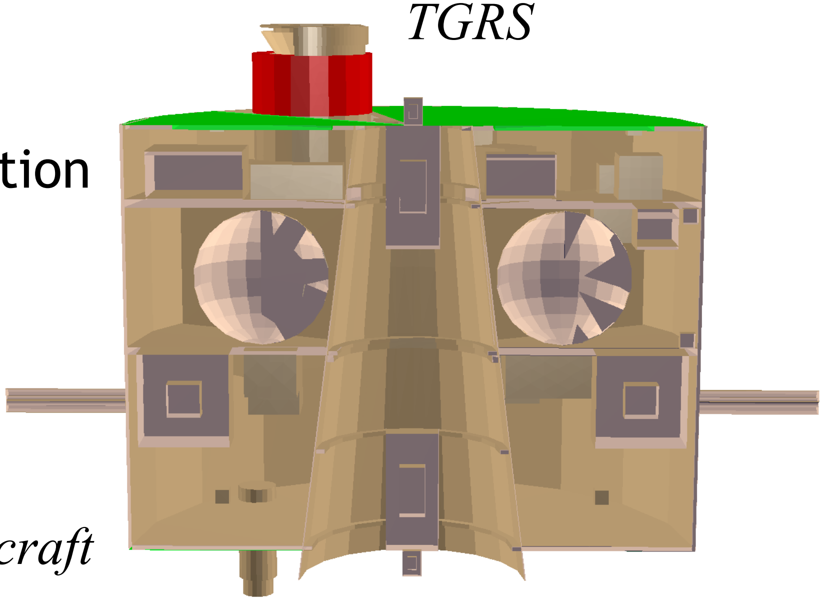
- Data reduction (e.g. Compton event reconstruction)
- Response generation (comparing the interaction information part and the reconstructed event parameters to generate response files)
- High level data analysis (e.g. image reconstruction)

Validation Example: TGRS

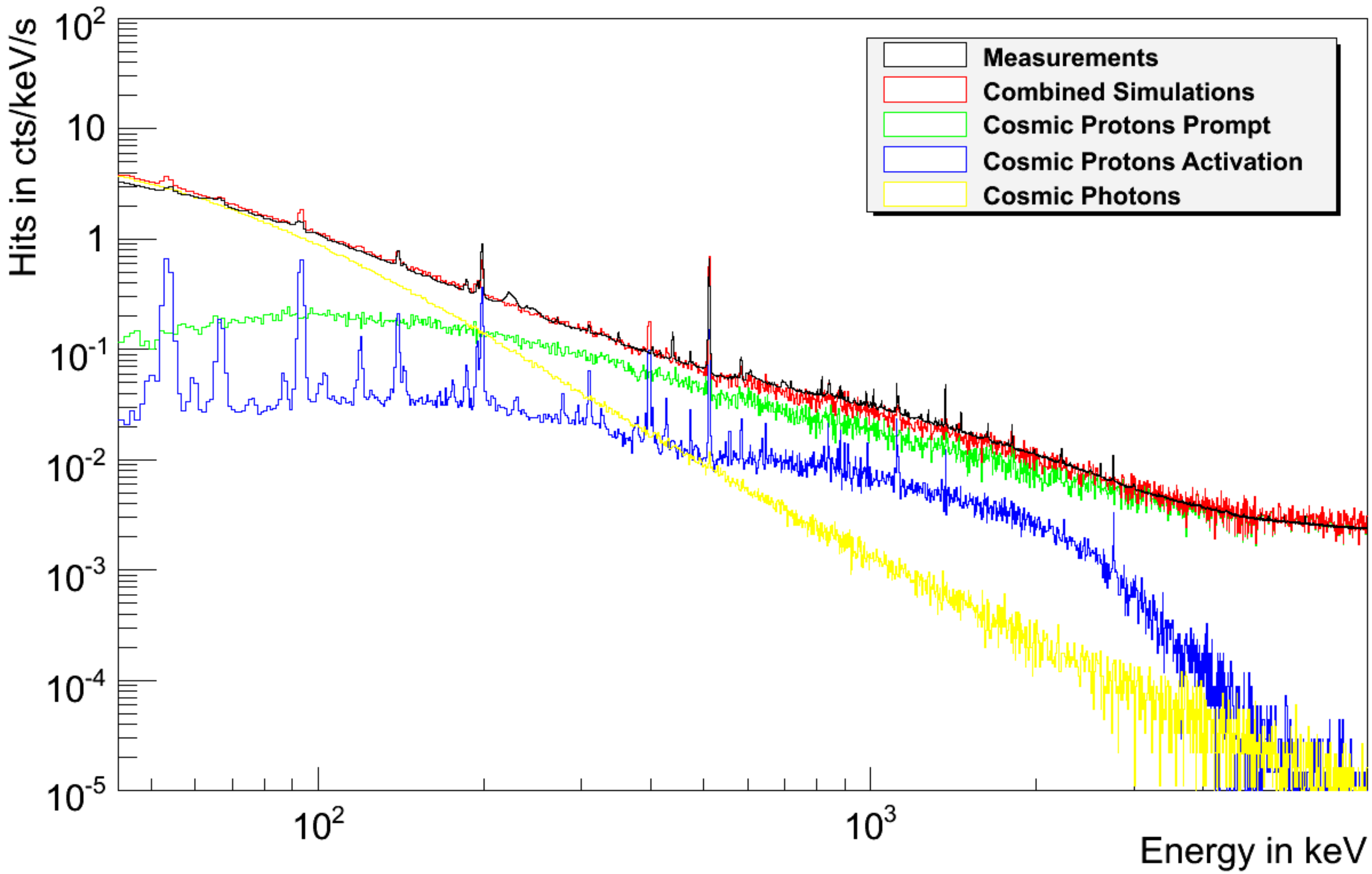
What is TGRS?

- ❖ TGRS (“Transient Gamma-ray Spectrometer”) is a Germanium-based gamma-ray detector aboard the Wind space craft
- ❖ Energy range: ~25 keV to ~8 MeV
- ❖ Designed for detection of transient gamma-ray sources (e.g. GRBs, solar flares)
- ❖ Position: Lagrange point L1
- ❖ Dominant background:
  - ➢ Cosmic photons
  - ➢ Cosmic protons + resulting activation

→ Nicely suited to validate activation simulations!



TGRS: Background simulations

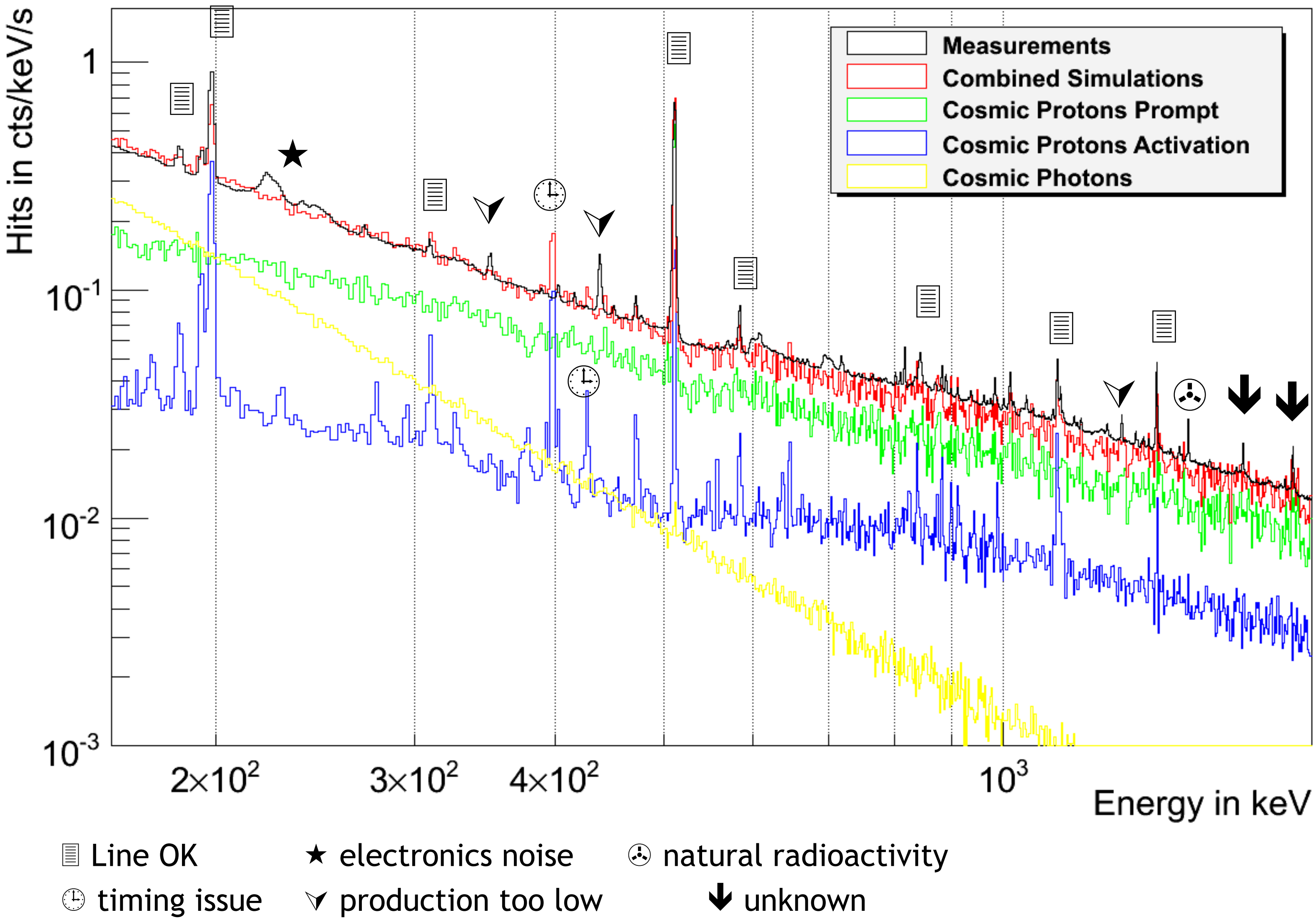


General shape in good agreement with measurement:

→ Slightly to high below ~60 keV (missed material in front of detector?)

→ Slightly to low between 0.6 and 1.5 MeV (missed activation component?)

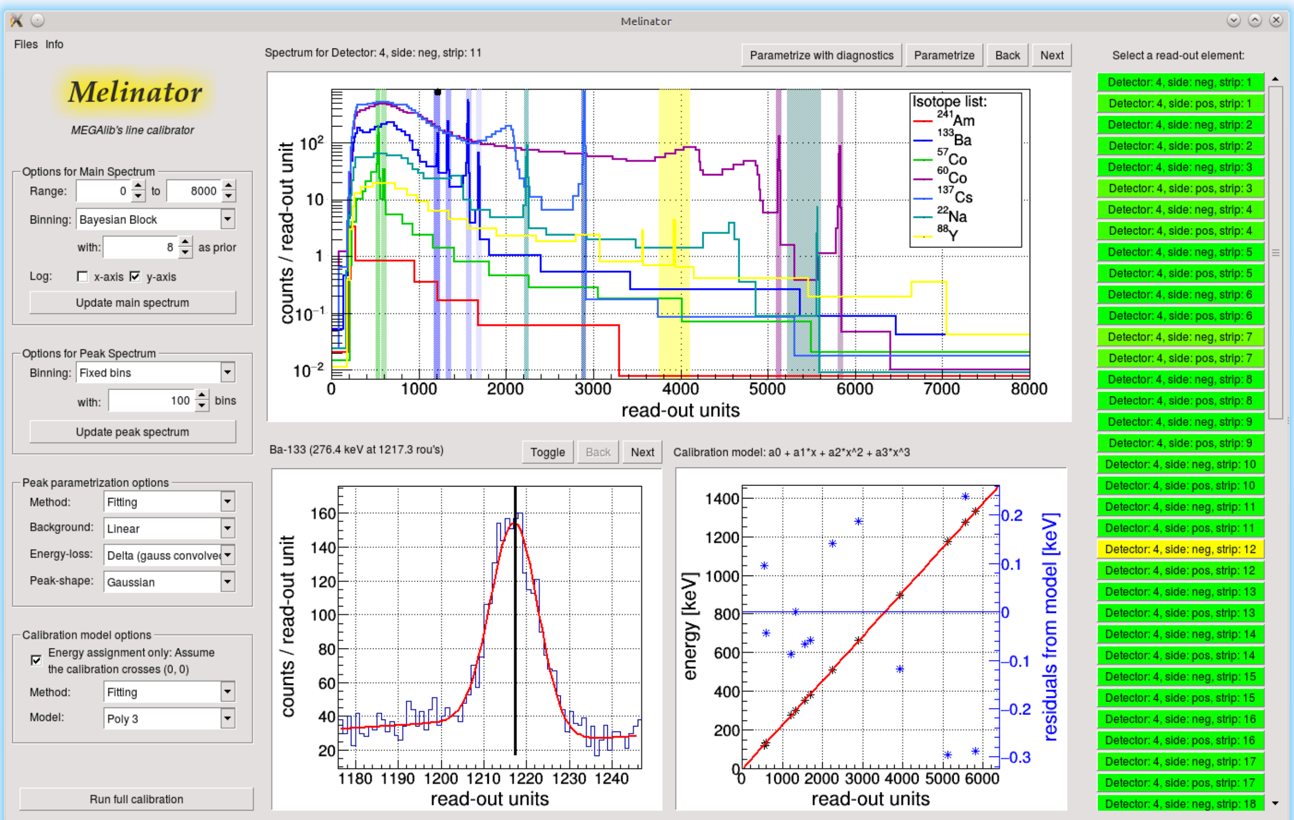
TGRS Simulations: Status of Various Lines



Core Tools Fretalon: Calibration

Fretalon is a calibration framework:

- Not a ready-to-use tool!
- User must add code to handle the given measurement system
- But many reusable modules available:



Melinator: MEGAlib line calibrator

The *.roa fretalon input file format

```
TYPE ROA

UF doublesidedstrip adcwithtiming

SE
ID 1

TI 1463450240.370016300

GX 153.03612 -9.0036867
GZ 250.02301 -37.513381
HX 346.62391 0.70627609
HZ 122.6425 89.018514

UH 1 3 p 87 90
UH 1 1 p 480 120
UH 1 2 p 248 130
UH 1 4 p 1127 180
UH 1 33 n 746 220
UH 1 31 n 236 135
UH 1 23 n 124 0
```

File format description (content of the UH section)

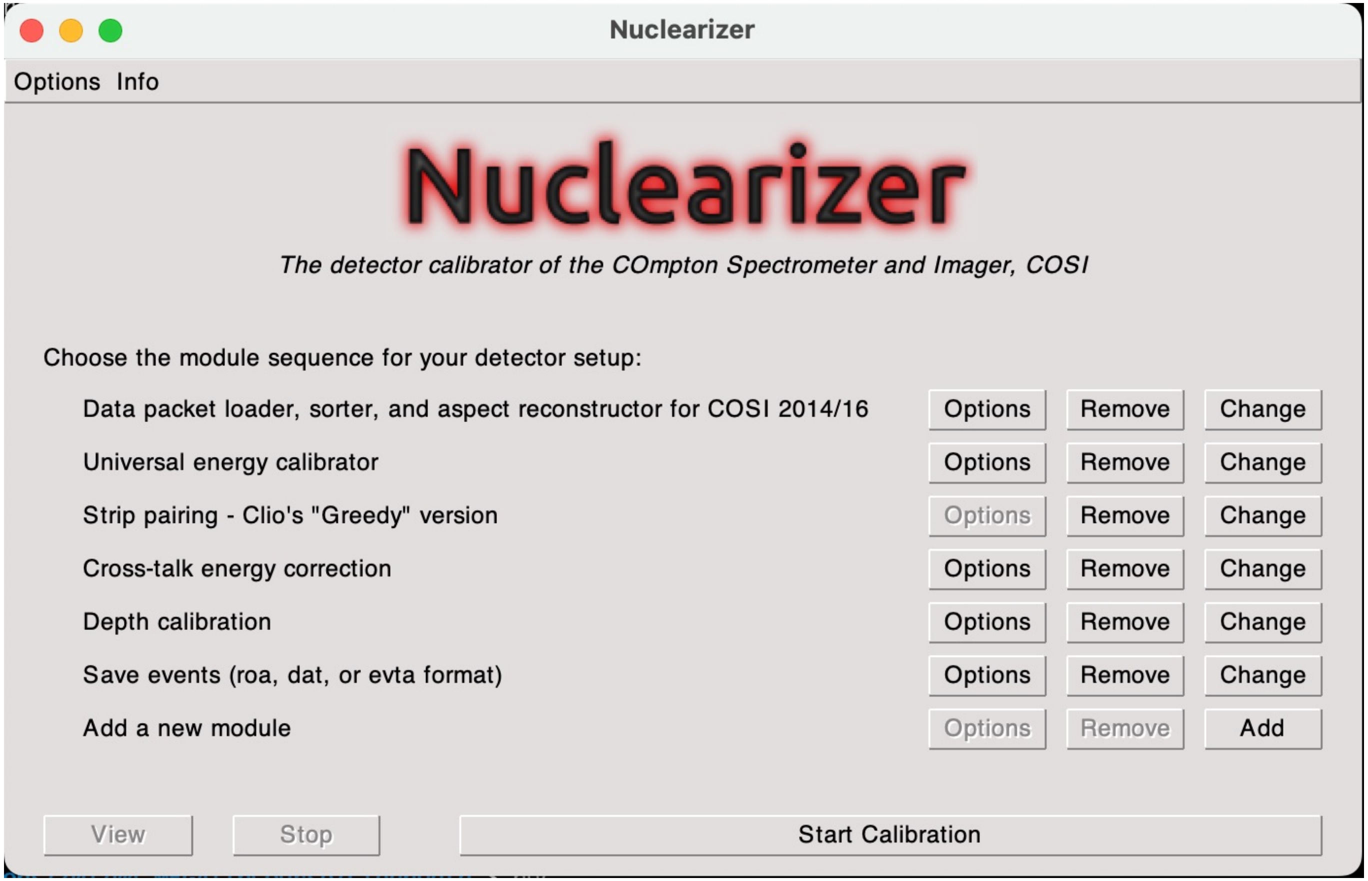
Event ID

Observation time

Instrument pointing in Galactic and horizon coordinates

Measurement data:
Uncalibrated strip hits with:
- Detector ID
- Strip ID
- Detector side
- AD-converter units
- Timing

Nuclearizer: Calibration



- Nuclearizer performs all steps in the calibration of COSI data.
- Designed to easily and independently create new calibration modules
- It is based on MEGAlib's fretalon framework.

The *.evta calibration output file format

```
SE

ID    4

TI 1465689986.707882700

GX 174.15588 -3.738098
GZ 261.16499 38.611258
HX 302.66145 -0.75092405
HZ 323.47109 89.196678

HT 3;-2.8184;-5.62935;-
   3.9399548;175.41029;0.057735027;0.057735027;0.032677695;0.63332153
HT 3;-0.8184;-6.82935;-
   5.321;161.18599;0.057735027;0.057735027;0.068554981;0.67390887
HT 3;-0.2184;-7.22935;-
   4.571;33.566996;0.057735027;0.057735027;0.4330127;0.61133573
HT 3;-3.0184;-5.42935;-
   4.0283986;145.54787;0.057735027;0.057735027;0.027560663;1.0655006
```

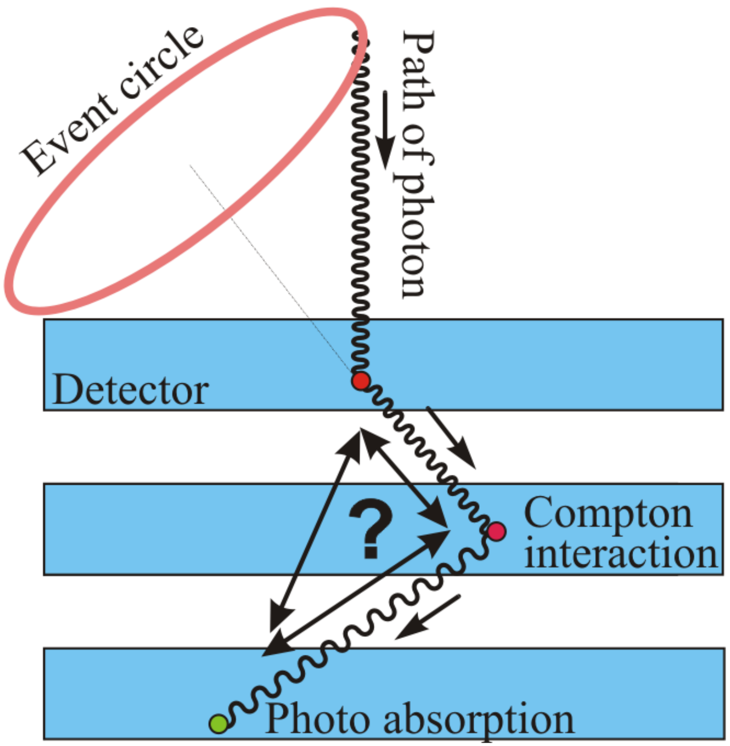
Event ID

Observation time

Instrument pointing in Galactic and horizon coordinates

Measurement data:
→ Positions and energy deposits and their uncertainties

Core tools Revan: Event Reconstruction



Typical Compton event:

The detector measures multiple interactions. The sequence of interactions is originally not known and has to be determined by event reconstruction.

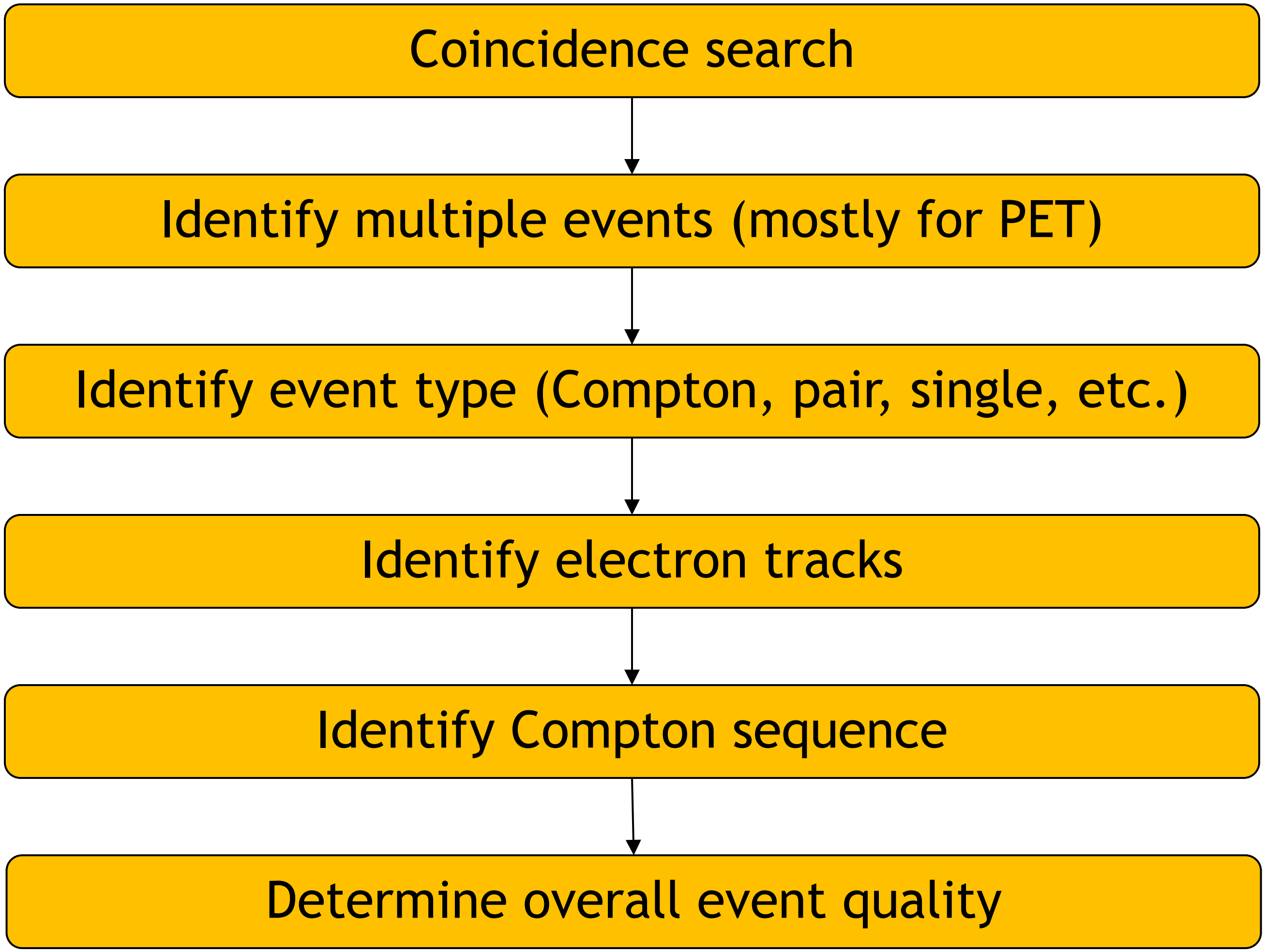
Main goals of event reconstruction:

- Reconstruct the path of the original photons and electrons
- Find the parameters of the original Compton or pair interaction
- Determine if the event originated from a completely absorbed non-background photon

Main challenges for astrophysics:

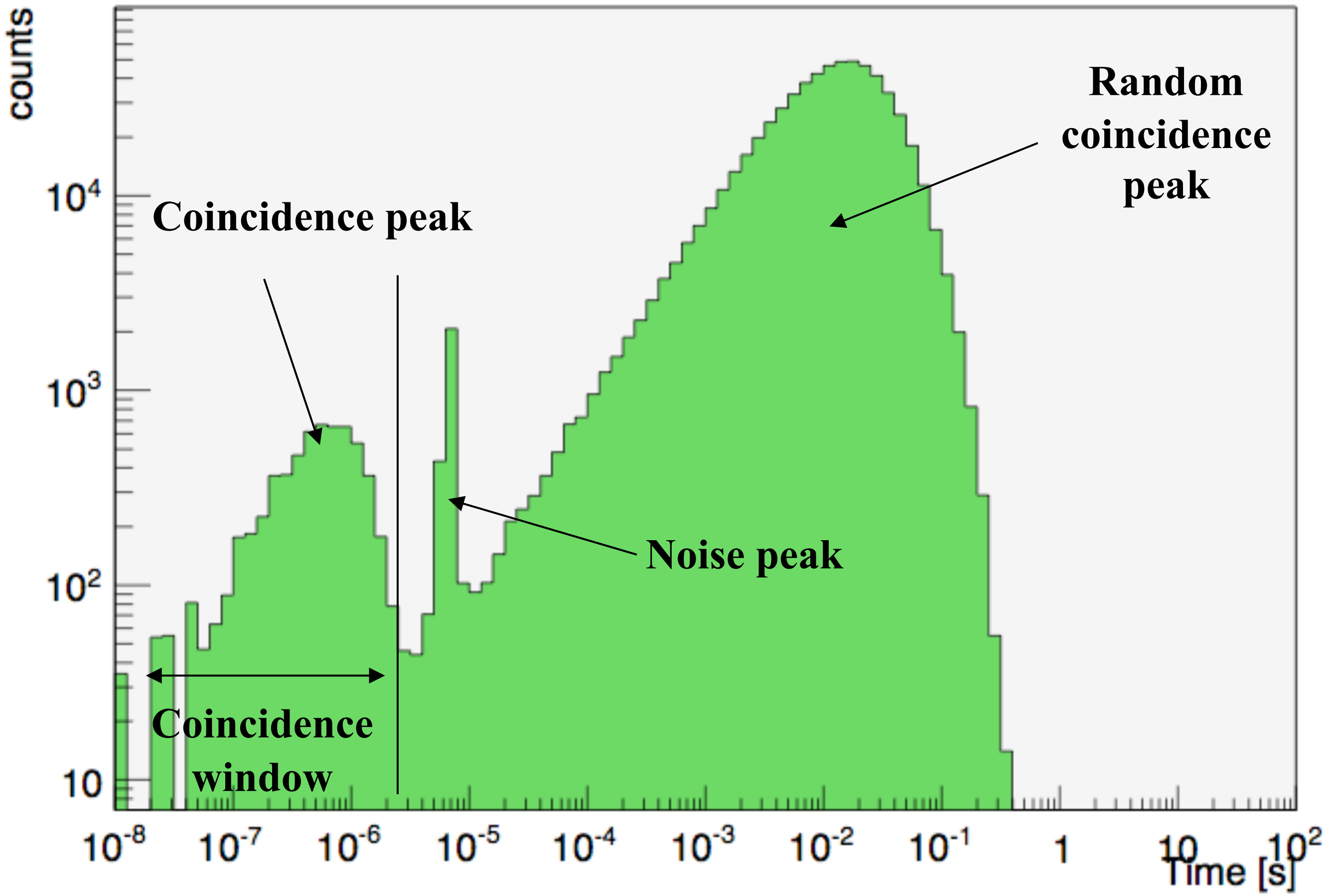
- Signal to background ratio up to 1:100
- Astrophysical nuclear lines (e.g. 511 keV, ^{26}Al) overlap with detector activation lines
- Hits in passive material

Event Reconstruction Steps:



Coincidence search

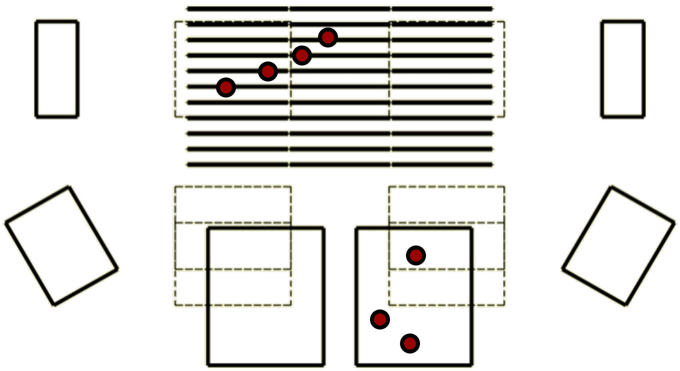
HEMI detector example:
Time distance between hits in different detectors



Combined hits into an event whose time distance is smaller than 2-3 μs

Event Reconstruction Steps

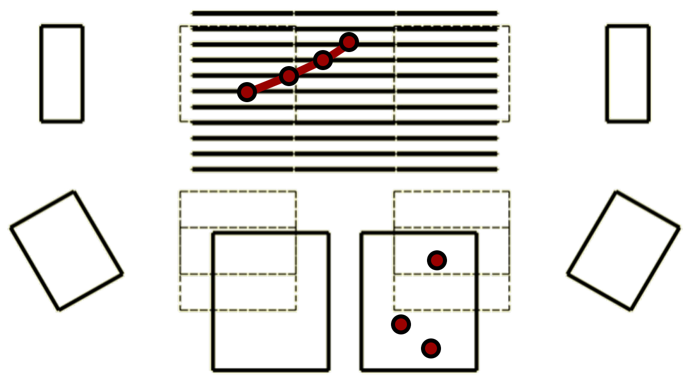
All of Revan's event reconstruction algorithms have the following analysis steps in common to find the correct ordering of the hits:



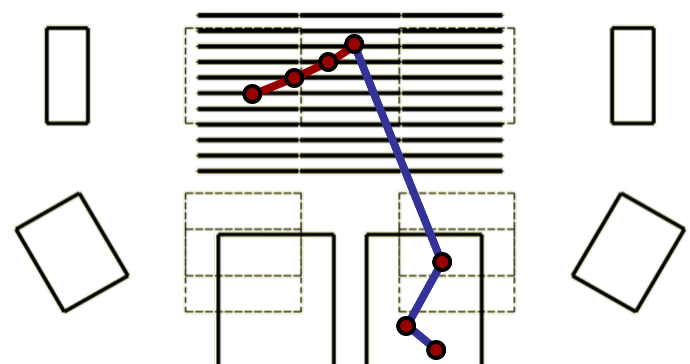
1) Start with a set of hits and first perform coincidence search and then clusterize coincident hits of adjacent voxels

2) Find charged particle tracks of e.g. Compton or pair events

Sketches of the MEGA detector drawn with the eview program



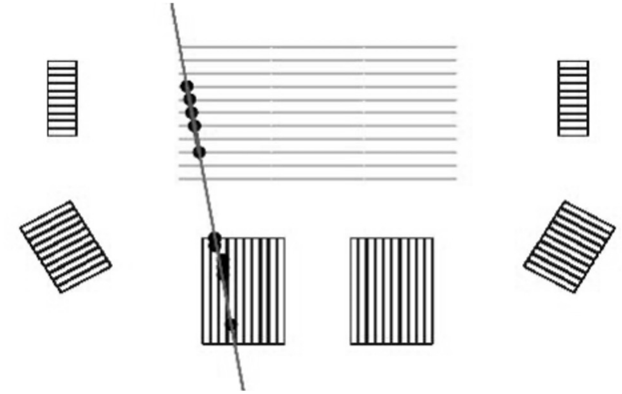
3) Finally find the overall Compton hit sequence and determine the quality of the event.



Electron tracking

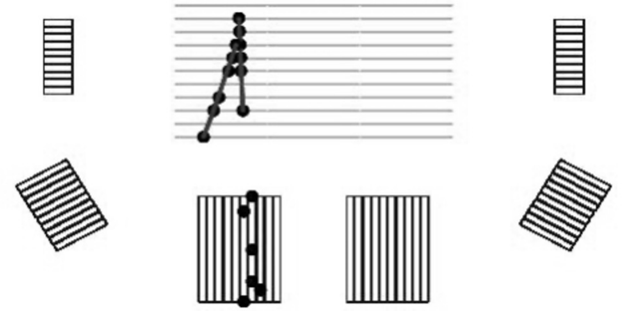
Step 1: MIPS/showers

- Find (almost) straight lines through your detector



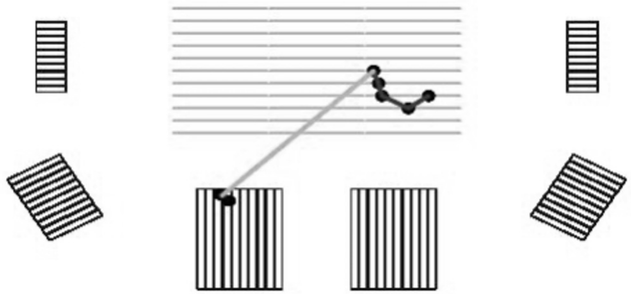
Step 2: Pair tracks

- Find Λ-shaped structures in your data set
- Find path into calorimeter



Step 3: Compton-electron tracks

- Find hits belonging to individual tracks
- Find the path
- Find the direction

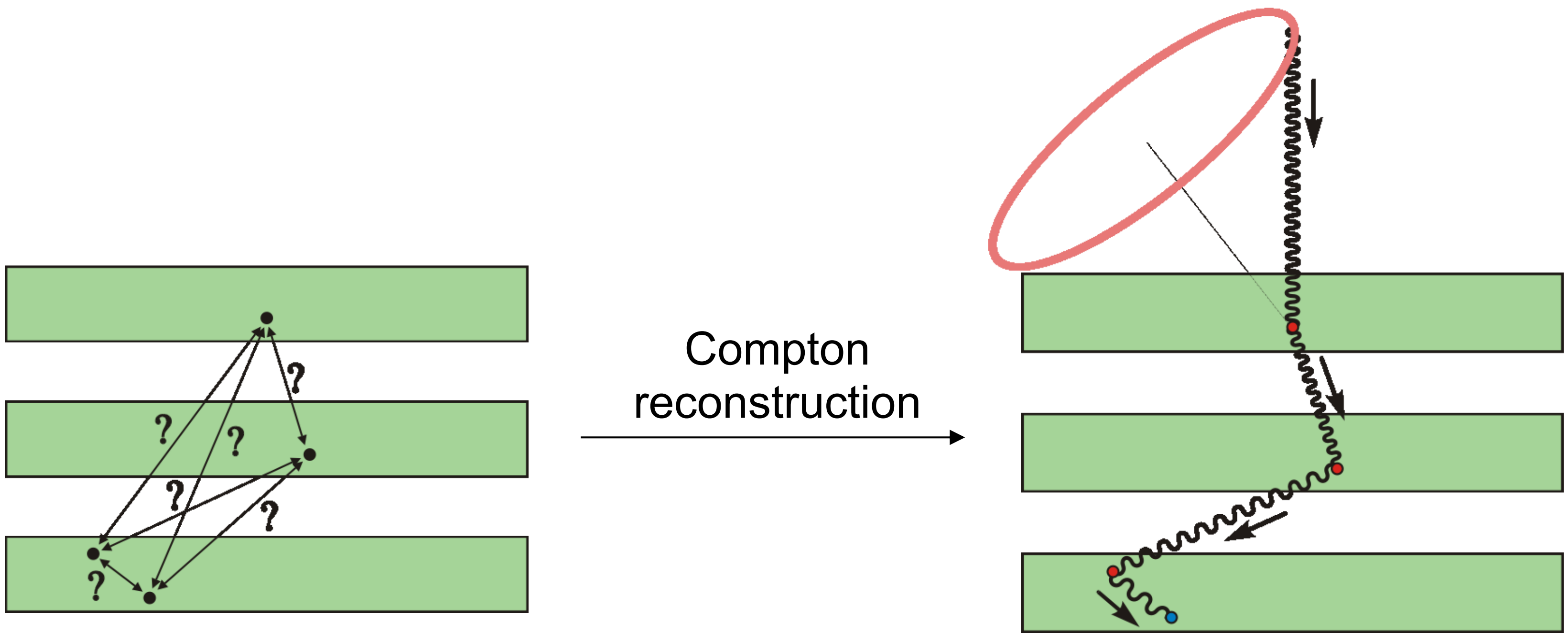


Compton Event Reconstruction

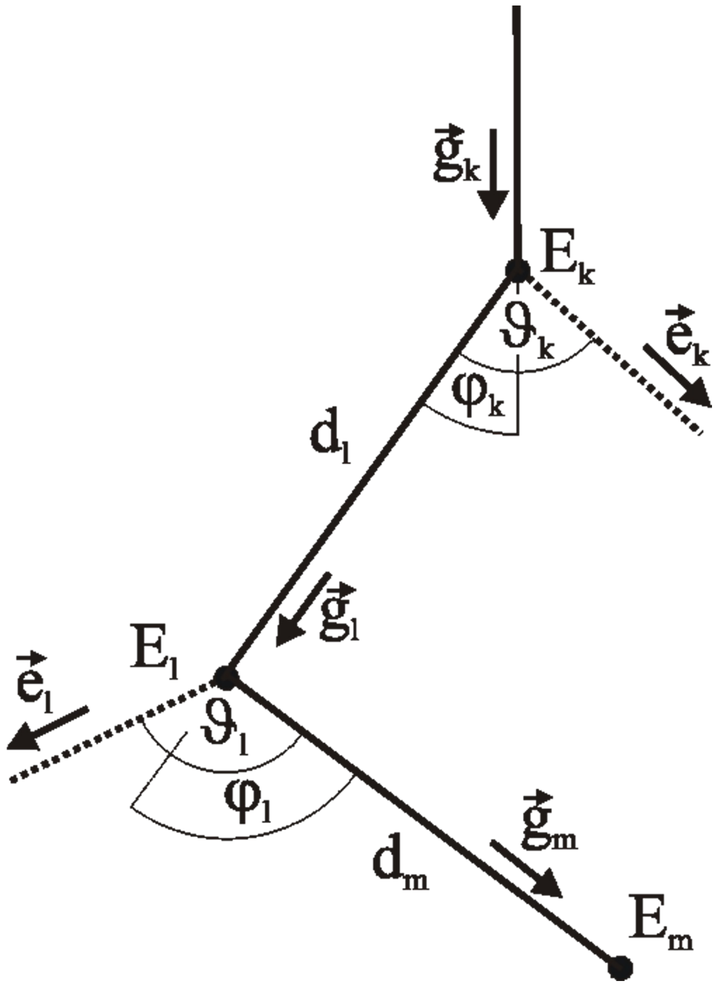
The Problem:

The detector only measures hits without time information

→ Path of photon is unknown!



Available Information



Basic data:

- All measured information: N x (x,y,z,E)

Enhanced data:

- Redundant scatter angles:

  Angles φ_l, ϑ_k, ϑ_l can be determined via geometry and via Compton kinematics (dφ, dϑ-criterion)!

- Absorption probabilities along d_l, d_m
- Klein-Nishina scatter probabilities
- Probabilities that the above are measured with the current geometry.

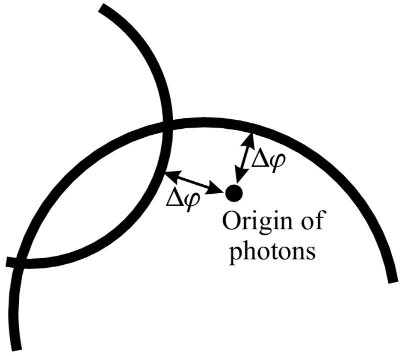
The Approaches

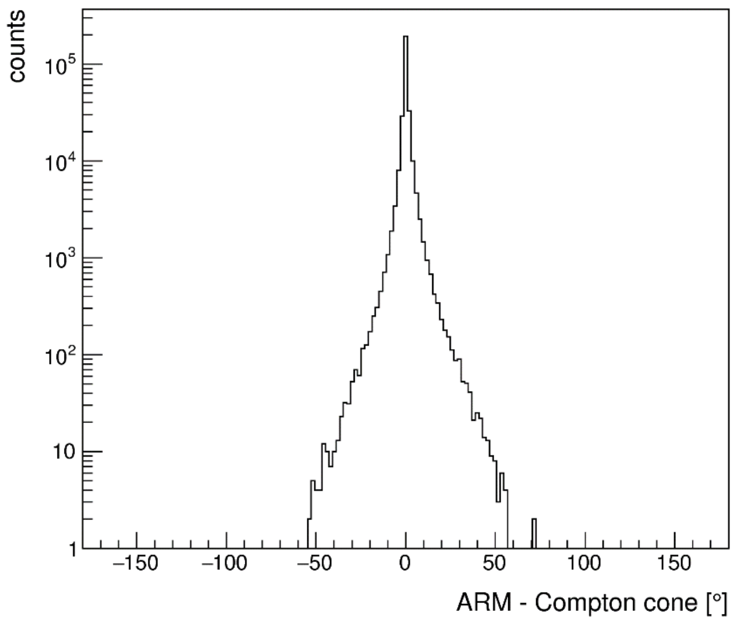
1. Classic approach
   - Only use the 1D projections of the complete data space
   - Real time but modest performance
2. Bayesian approach
   - Create a large data space with the most valuable dimensions
   - Slow, but good performance and best background rejection
3. Random forest of boosted decision trees
   - Use all measured data directly
   - Fast to train, fast to apply,
4. Neural net approach
   - 2-layer MLP
   - Best performance, but hard to fully validate

Event Reconstruction Performance Metric: ARM

ARM:
Angular Resolution Measure

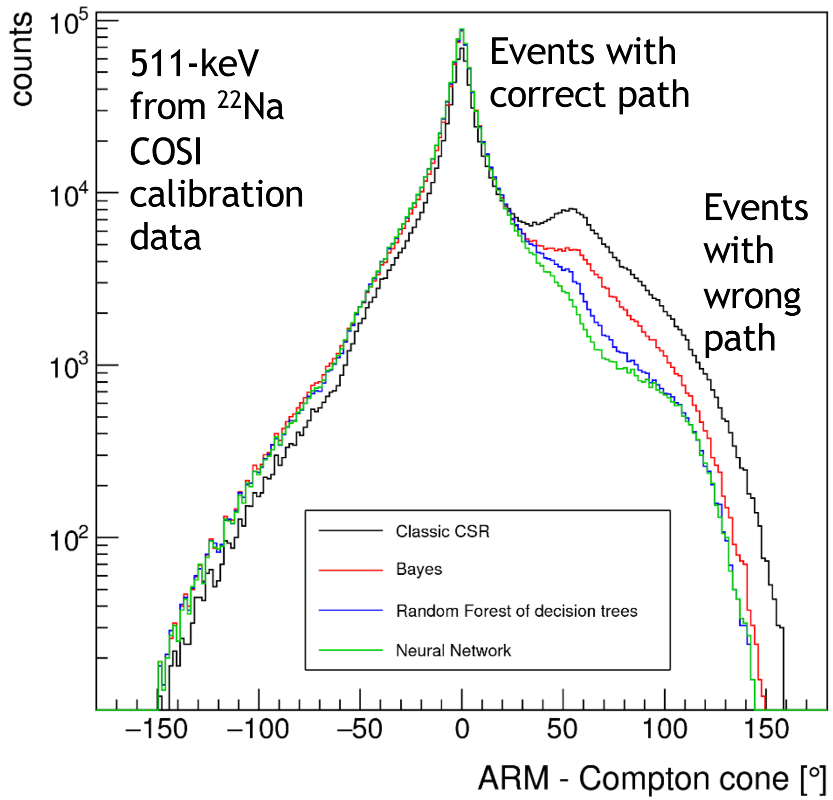
Smallest distance between Compton cone and known origin of gamma rays





ARM of ideal detector: Only Doppler-broadening
Reality: broadening by detector position and energy resolution

Event Reconstruction Performance



RMS improvement:

CSR: 0%

Bayes: 17.0%

RF-BDT: 24.9%

NN: 27.4%

Why is neural network best:

- Accepts all possible data (same as random forest)
- Best at learning non-linearities

Output of Revan

```
SE

ET CO
```
Event type: PH=Photo effect (single site), CO: Compton

```
ID 13
```
Event ID (number)

```
TI 0.328531810
```
Time

```
PQ 0
CT 0 1
TQ 0 0
```
Quality factors

```
SQ 2
```
Compton sequence length

```
TL 1
TE 135.363
```
Track length and initial deposit

```
CE 291.406 2.15617   135.363 3.25392
CD -3.5 4.9 -7.5   0.288675 0.288675 0.288675   0.7 1.4 0.0
0.288675 0.288675 0.288675 0 0 0   0 0 0
```
Compton scattering parameters, energies, positions, track directions as well as uncertainties

```
LA 9.61821
```
Minimum distance between any hits in the sequence

Response Generator

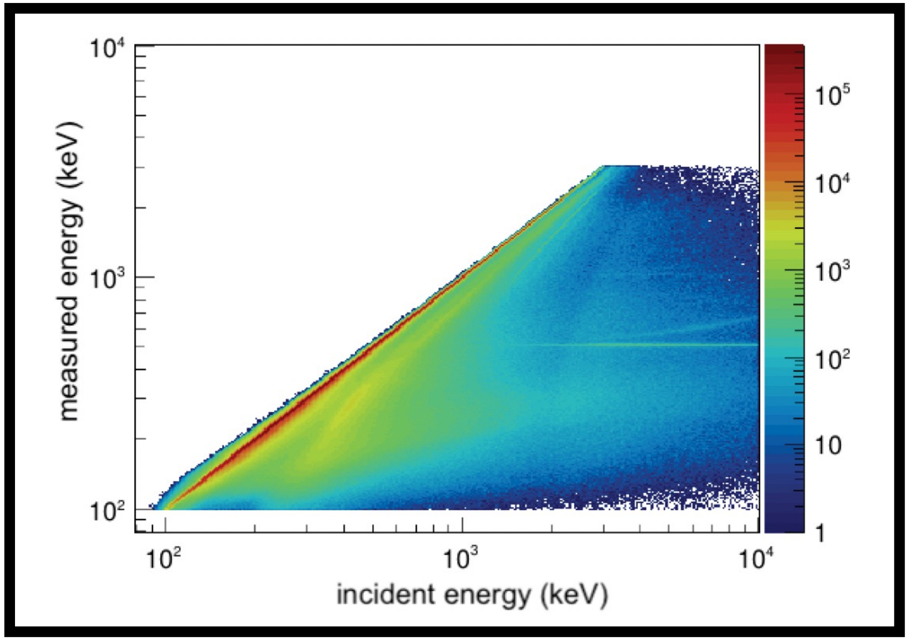
High level data analysis requires a detailed description of the detector response to the incident gamma-rays. This program compares

- Noised & discretized hit information, which resemble the real measurements

with

- Ideal interaction information

to retrieve the relation between initial and measured photon parameters to generate response matrices for event reconstruction and imaging.



COSI spectral redistribution matrix, the energy response

Responses & machine learning

Creates responses for:

- imaging:
  - list-mode
  - binned-mode far field
  - binned-mode near field
- spectral analysis
- polarization analysis
- and several more

Performs machine learning tasks

- cleaning training data for machine learning approaches
- training of the machine learning approaches:
  - CSR: Naïve Bayesian, random forest, neural-network
  - Electron tracking
  - Event identification

Core Tools: Mimrec

Mimrec provides:

- Event selections of all performance-relevant parameters of Compton and pair telescopes/cameras
- List-mode likelihood image reconstruction in spherical as well as Cartesian coordinates (2D, 3D) including different response calculation approaches for Compton and pair creation events
- General detector performance analysis (angular resolution, energy dispersion, scatter angle distributions, etc.)
- Performance assessment of event reconstruction algorithms
- Background corrected polarization analysis
- Sensitivity and background calculation tools
- … and many more

Event selections

Goal:

Optimize signal to noise ratio, i.e., improve the sensitivity of the instrument

- Throw out regions in your data space where background or bad events accumulate
- But requires careful analysis of effectiveness of cuts!

Most common cuts for astrophysics:

- Earth horizon cut: Eliminate events whose Compton cones intersect Earth's horizon
- Distance cuts: Set a minimum distance between the first two interaction positions → Improved angular resolution
- Energy cuts: Exclude background lines, e.g., 511 keV
- Quality factors during event selections
- Compton scatter angles (e.g. large scatter angles = high Doppler broadening)

Mimrec's Imaging Features

- Event types: Compton (tracked and untracked events) and pair events, and single sites events with coded masks
- LM-(OS-)ML-EM based with different response optimizations
- Coordinate systems: Cartesian 2D & 3D, spherical and Galactic coordinate systems
- Multi-threaded
- Speed-optimized (fast file parsing, exchange costly maths functions with approximations)
- Memory optimized (1-byte, sparse image storage)
- Create animations

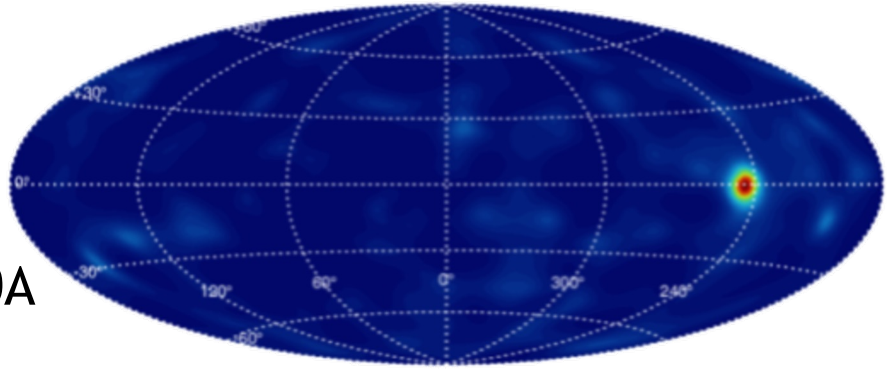
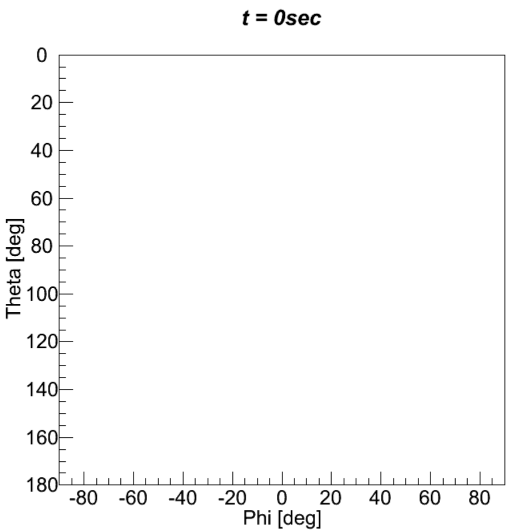
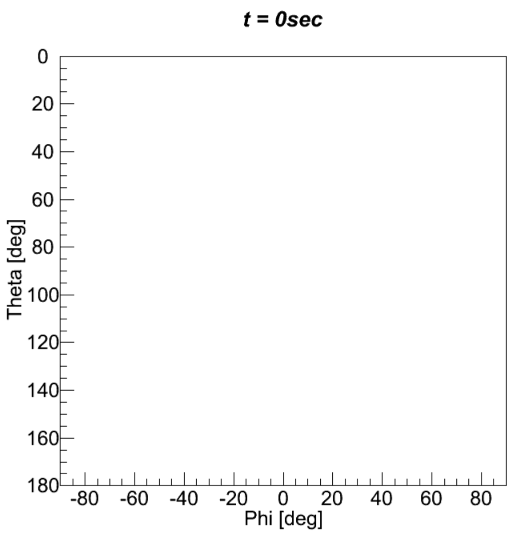


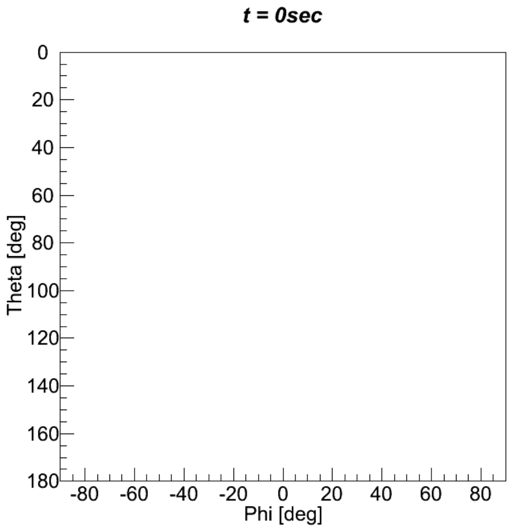
Image of GRB 20160530A measured with COSI

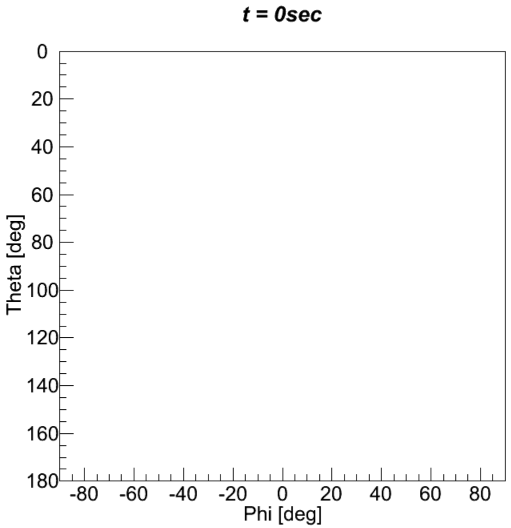
Animations

Real HEMI measurements









Imaging Examples

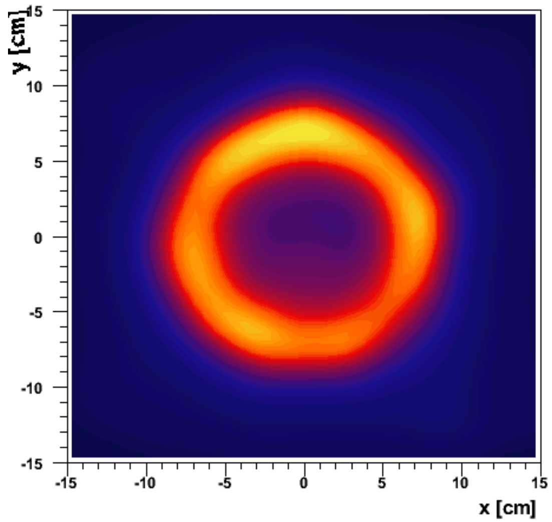
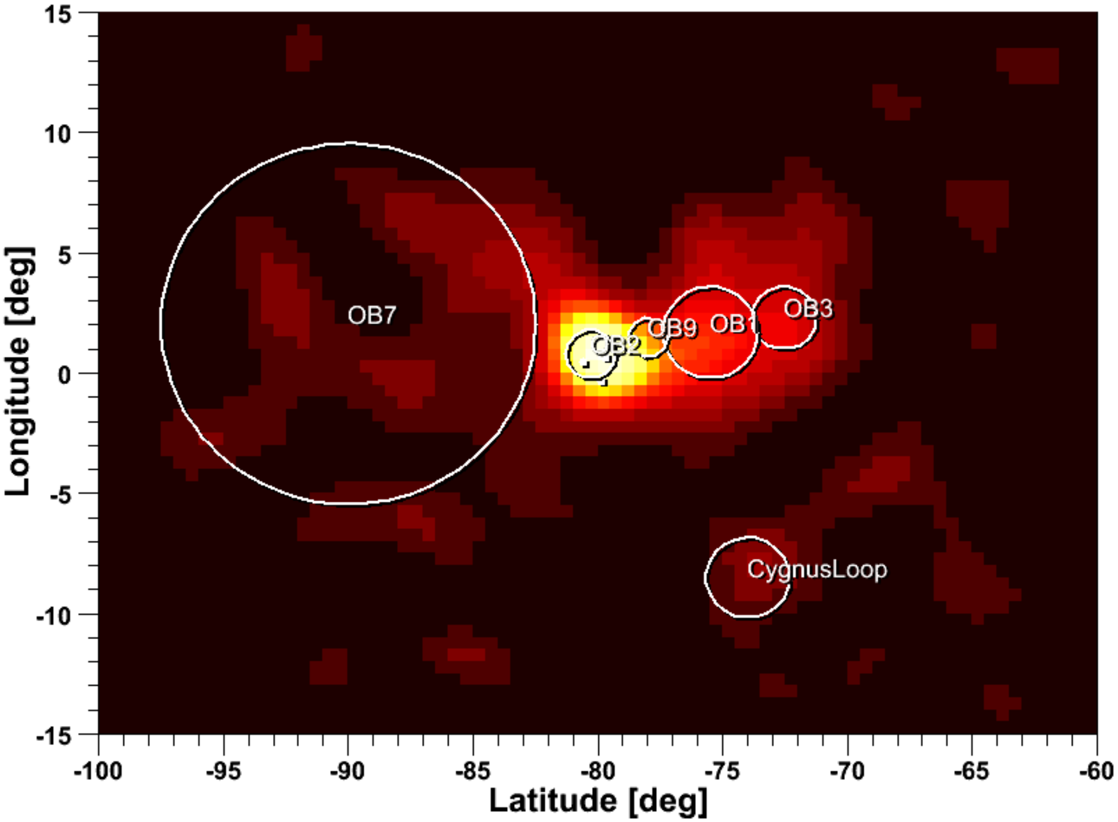


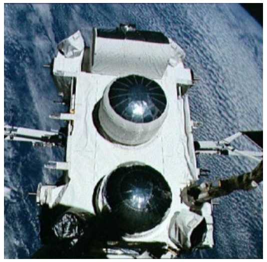
Illustration of the LM-EM algorithm recovering the image of a ring (Real measurement with MEGA prototype)

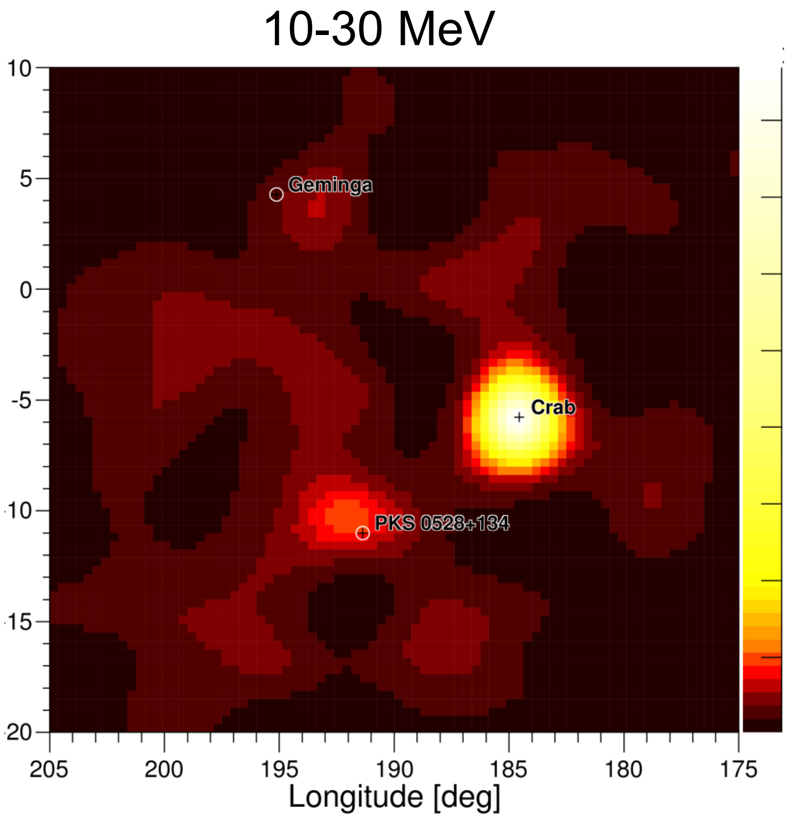
Image of the Cygnus region in the light of ^{26}Al after 2 years exposure (MEGA detector)

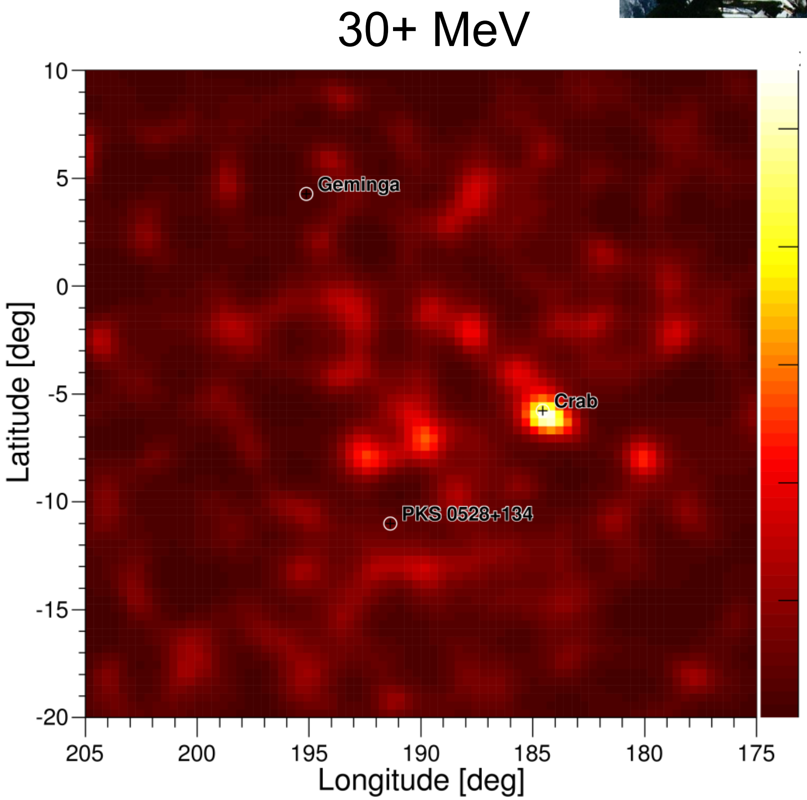


Galactic Anti Center With COMPTEL

Measurements with the COMPTEL space telescope (Compton events, data from 1991-1997):



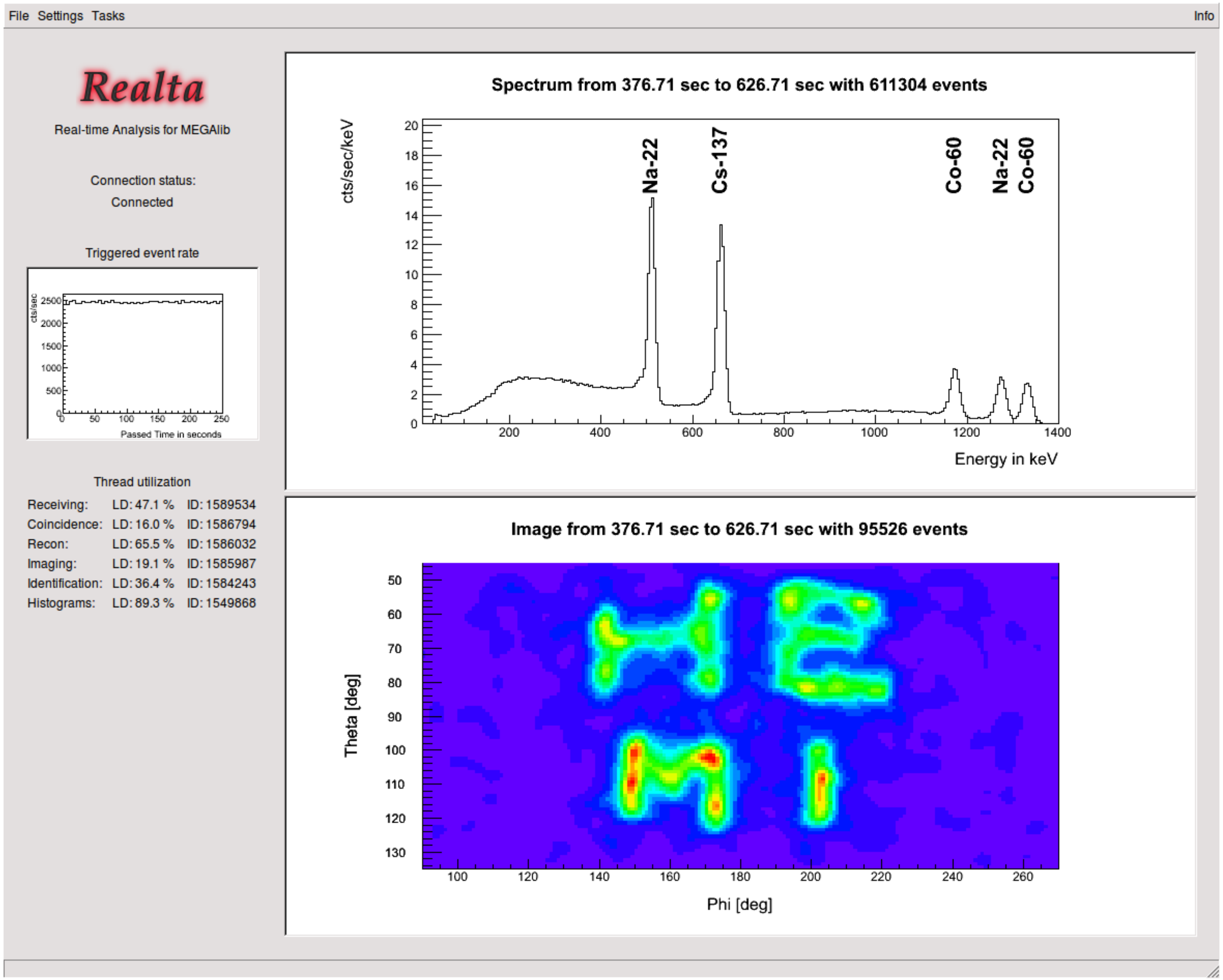




Source: Crab (pulsar), PKS 0528+134 (blazar)

Realta - Real-Time Analyzer

Real-time event & image reconstruction



Realta workflow

Goal is to provide real-time measurement analysis:

- Check if detector system works and measurement setup is OK
- Real-time nuclear threat detection

Makes heavy use of all other MEGAlib APIs:

- Geomega
- Revan
- Spectralyzer
- Mimrec

Data acquisition – here via a TCP/IP connection from the hardware or from a program reading simulated data

↓

Coincidence determination within a given time window

↓

Event reconstruction of Compton, pair-creation, and charged particle events

↓

Spectral identification of isotopes by looking at detected nuclear lines

↓

List-mode image reconstruction in near or far field

↓

Display of the final results

Other tools / libraries

Eview

- Combines Geomega and Revan to display reconstructed events

Spectralyzer

- Spectral analysis and isotope identification library

SensitivityOptimizer

- Uses background and source simulations to optimize event selections for finding the best sensitivity of a Compton and/or pair telescope

Enhancements for version 4

Version 4 will include all the enhancements for analysis of the COSI SMEX mission:

- Simulations: More spectral and beam inputs for COSI data challenge
- Geomega: CAD import
- New more detailed detector effects engine
- Response creation overhaul and new Compton responses
- New strip pairing approach
- Improved machine-learning-based event reconstruction
- Background identification
- Improved imaging
- Full documentation
- Full set of unit tests
- Many new diagnostics functions

- All improvements will show up in version 4 from now till 2026
- Main branch is always a working version

Some References

MEGAlib in general:

A. Zoglauer et al., "*MEGAlib – the Medium Energy Gamma-ray Astronomy library*", NewAR 50, p 629-632, 2006

Geomega/Cosima:

A. Zoglauer et al., Cosima - The cosmic simulator of MEGAlib, NSS Conference record, 2009

Revan:

A. Zoglauer, "*First Light for the Next Generation of Compton and Pair Telescopes*", Doctoral thesis, TU Munich, 2005

Mimrec:

A. Zoglauer et al., "Design, Implementation, and Optimization of MEGAlib's image reconstruction tool Mimrec", NIM A 652, 2011

Realta:

A. Zoglauer et al., "Status of MEGAlib's Real-Time Analysis Tool Realta", NSS Conf. Rec. 2011

Spectralizer:

M. Galloway et al., "Spectral Analysis for the High-efficiency Multimode Imager", NSS Conf. Rec. 2010