The MEGAlib software package

written by Andreas Zoglauer

1 Preamble

MEGAlib stands for Medium Energy Gamma-ray Astronomy Library. It represents the complete data analysis chain from simulations to high-level data analysis like image reconstruction in the Compton scattering energy regime. It has been originally developed for the MEGA telescope.
Although the library has undergone thorough testing and although there is no part of the program, which has not already been successfully used, like all other programs, also MEGAlib is not free of errors. Since the development of MEGAlib was only intended to get the work done I need for my research, MEGAlib will probably never leave its development phase. However, especially the older parts of the software are now in use for several years and can be considered as reliable and stable. In order to minimize the number of bugs and thus to improve the performance of the library and all its programs, please report all crashes, incompatibilities and other bugs to the following email address:

zog@ssl.berkeley.edu

Moreover if you think some features are missing or if you have enhancement requests, feel free to let me know — but consider, I am astrophysicist, no software developper!

Andreas Zoglauer
Berkeley, June 2006

2 Installation

Currently the platforms Linux, Solaris and MacOS X are supported. No other Unix versions have been tested. The programs will definitely not run under Windows.
A detailed description of the installation can be found in the file INSTALL.txt, located in $(MEGALIB)/doc.

3 License

The license is located in $(MEGALIB)/doc/License.txt. Please read it carefully since it contains important information.

4 Special Options

4.1 Main Makefile

gmake	Compile all programs
gmake clean	Remove all *.o, *.so files
gmake <program name>	Compile this specific program and launch it
gmake clean_<program name>	Clean all files related to the program
gmake texdoc	Tex this documentation and launch gv
gmake man	Generate doxygen documentation and place it in doc/html/index.html

4.2 Special scripts

launch <program name> <options>	Recompile (if necessary) and launch the given MEGAlib program from every directory
debug <program name> <options>	Recompile (if necessary) and launch the given MEGAlib program from every directory in debugging mode

5 Program overview

This is intended as a brief overview over the different programs which are available and thus the tasks which can be performed with MEGAlib.

5.1 Geomega

Geomega is the detector geometry package. All geometry and detector information is represented by a *.geo.setup file. This includes the description of the volumes and materials as well as the detector properties. The geometry can be converted to formats which the GEANT3 simulation program GMega as well as MGeant/MGGPOD can process. The Geant4 program Cosima directly loads Geomega's shared library to use the geometry package. Almost all other programs use this geometry for geometry display (eview) as well as event and image reconstruction (especially absorption probabilities and resolutions).

5.2 GMega

Simulation tool based on Geant3. It is capable of handling different input spectra and source geometries. For details see the documentation in the GMega directory. It produces an output file, which can be read from the event analysis programs, like Sivan or Revan.

5.3 Cosima

Simulation tool based on Geant4. Like GMega it is capable of handling different input spectra and source geometries. For details see the documentation in the Cosima directory. It produces an output file, which can be read from the event analysis programs, like Sivan or Revan.

5.4 Sivan

Sivan is intended for the analysis of simulated data and explicitly using simulation information: the underlying library is mainly used to determine responses and determine the maximum possible efficiency of an event reconstruction algorithm.

5.5 Revan

The programs provides the event reconstruction, i.e. it handles the transition of the events from pure hits into interaction processes (Compton, Pair, etc.). The algorithm is split into 4 subsections: clustering (blobbing adjacent hits into one larger hit), tracking (finding showers, muons, pair events and Compton electron tracks), Compton sequence reconstruction (identifying the sequence of Compton interactions) and finally decay detection (identifying events, which might originate from decays).

5.6 Mimrec

Mimrecs main duty is list-mode likelihood image reconstruction. In addition it can handle other high-level data analysis, like compiling spectra, ARM-distributions, scatter-angle distribution, and more.

5.7 ResponseGenerator

Program to generate responses/data-space used for the Bayesian event reconstruction algorithms

5.8 ResponseManipulator

Merge and view response files

5.9 ConvertMGGPOD

Converts fits files generated by MGeant/MGGPOD into the SIM format, which can be read by the event reconstruction programs

5.10 SensitivityOptimizer

Calculates the sensitivity based on a source file and several background files in a multi-dimensional data space

6 File formats

All file formats have some header information, which include the type of the file, its version, and the geometry information:

The type of the file
Type <Type $\in \{SIM, TRA, etc.\}$>

The version of the file
Version <Version number as int>

The geometry
Geometry <Path to the geometry>

6.1 The SIM format

Start event
SE

Event ID
ID <ID triggered event> <ID simulated event>

Time in seconds Linux time
TI <Time in sec>

Total deposited energy
ED <Energy in keV>

Energy deposited in not sensitive material
NS <Energy in keV>

Energy deposited in the guard ring of strip detectors
GR <x position of detector center in cm>
<y position of detector center in cm>
<z position of detector center in cm>
<Energy in keV>

Total energy deposit per drift chamber
XE <x position of interaction in cm>
<y position of interaction in cm>
<z position of interaction in cm>
<Energy in keV>

Electron direction information for Strip3DDirectional detectors

DR <x position of interaction in cm>
<y position of interaction in cm>
<z position of interaction in cm>
<x direction>
<y direction>
<z direction>
<Energy in keV>

Interaction information version 3

IA <Detector ID 1=2D strip, 2=MEGA calorimeter, 3=3D strip, 4=ACS, 5=drift chamber>
<Type $\in \{INIT, PAIR, COMP, PHOT, BREM, RAYL\}$>
<Particle number 1: Photon, 2: Positron, 3: Electron>
<ID of this interaction>
<ID of interaction this particle originated from>
<x position of interaction in cm>
<y position of interaction in cm>
<z position of interaction in cm>
<x direction of new particle>
<y direction of new particle>
<z direction of new particle>
<Kinetic energy of new particle in keV>

Interaction information version 22

IA <Type $\in \{INIT, PAIR, COMP, PHOT, BREM, RAYL\}$>
<ID of this interaction>
<ID of interaction this particle originated from>
<Detector ID 1=2D strip, 2=MEGA calorimeter, 3=3D strip, 4=ACS, 5=drift chamber>
<x position of interaction in cm>
<y position of interaction in cm>
<z position of interaction in cm>
<Mother particle number 1: Photon, 2: Positron, 3: Electron>
<x direction of original particle>
<y direction of original particle>
<z direction of original particle>
<x polarization of original particle>
<y polarization of original particle>
<z polarization of original particle>
<Kinetic energy of original particle in keV>
<Particle number 1: Photon, 2: Positron, 3: Electron>
<x direction of new particle>
<y direction of new particle>
<z direction of new particle>
<x polarization of new particle>
<y polarization of new particle>
<z polarization of new particle>
<Interaction time in sec> *(Missing in Version 21)*
<Kinetic energy of new particle in keV>

Hits

HT <Detector ID 1=D1, 2=D2, 3=D3, 4=ACS, 5=drift chamber>

<x position of interaction in cm>
<y position of interaction in cm>
<z position of interaction in cm>
<Energy deposit in keV>
<Time since start of event>
<Vector of IDs of the interactions, which contributed to this hit>

6.2 The GEO.SETUP format

All parameters of the MEGA detector are initialized in a setup file. Its characteristic suffix is "setup". It has an object oriented, keyword based style, for example in the beginning a detector is declared and afterwards it is initialized with several parameters.

```
MSiStrip Poseidon
Poseidon.Threshold 0.0 0.0 0.0 100.0 100.0 100.0

Volume WorldVolume
WorldVolume.Material Germanium
WorldVolume.Shape BRIK 0.0 0.0 0.0
```

A complete list of all keywords can be found in chapter keywords.
In order to assure some kind of safeness the setup file is scanned four times:

1. Include all "Include"-files
2. Scan for other command keywords and object keywords
3. Scan for Clones of the object keywords
4. Assign all other parameters
5. Validate the input

This assures that the sequence of keywords in the file can be arbitrarily.
All dimensions of the variables are in cm, keV, g, etc.
There exist three types of keywords: Command keywords, object keywords and parameter keywords. These keywords and the whole setup file are not case sensitive.
There are several important limitations with this geometry and detector description format compared to Geant3 and Geant4:

1. All daughter volumes have to be completely contained in their mother volumes.
2. No overlapping volumes are allowed (with exception of virtual volumes)
3. There are only a limited number of volume types implemented so far. If you need more, let me know, but make sure, that they exist in Geant3, Geant4 and Root.
4. It is not possible to divide one volume in sub volumes.
5. All detector volumes need to be boxes (with exception of the Scintillator- and Anger Camera-type detectors)
6. ... and probably many more ...

6.2.1 Global keywords

Keyword	**Name** <text>
Description	Gives the setup a distinguishable name
Example	`Name MegaPrototype`

Keyword	**Version** <float>
Description	Version of the current setup
Example	`Version 1.01`

Keyword	**Include** <file name>
Description	Include the file <file name> into the setup-file. This is very useful when splitting a large setup-file into smaller parts, e.g. defining an own file for each detector or a file containing the specific materials, etc. The file name is allowed to be a relative path and might contain wildcards
Example	`Include $(MEGALIB)/resource/geometries/materials/Materials.geo`

Keyword	**SurroundingSphere** <radius> <x-position> <y-position> <z-position> <distance>
Description	A surrounding sphere is a mandatory requirement for the simulations with GMega or Cosima. If you do simulations in the far-field, the photons are simulated from a disk with radius <radius> pointing from a distance <distance> towards point <x-position> <y-position> <z-position>. This feigns a real far-field. Therfore, make sure you define a surrounding sphere which is as small as possible but encloses the complete detector geometry, without intersecting any volume.
Example	`SurroundingSphere 25.0 0.0 0.0 15.0 25.0`

Keyword	// <text> or # <text>
Description	Comment
Example	`# This is a comment`

6.2.2 The object Material

Keyword	**AbsorptionFileDirectory** <path>
Description	Gives a default directory where to search the absorption files
Example	`AbsorptionFileDirectory $(MEGALIB)/resource/geometries/materials`

Keyword	**Material** <text>
Description	Defines the material <text>
Example	`Material CsI`

Keyword	<Material>.**Density** <float>
Description	The density of the volume in g/cm^3
Example	`CsI.Density 4.5`

Keyword	<Material>.**ComponentByAtoms** <A> <Z> <Number of atoms of the component/molecule)>
Description	Declares a component of a material. For example if the material is a mixture of the two components Cs and I, they have to be declared as in the example below. The number of atoms needs to be an integer!
Example	`CsI.Component 132.9 55 1 // Cs` `CsI.Component 126.7 53 1 // I`
Keyword	<Material>.**ComponentByMass** <A> <Z> <Fractional mass of the component in the material>

Description	Declares a component of a material. For example if the material is a mixture of the two components N and O, they have to be declared as in the example below. The total fractional mass needs to be 1!
Example	`Air.Component 14.0 7.0 0.7 // N` `Air.Component 16.0 8.0 0.3 // O`

Keyword	<Material>.**RadiationLength** <float>
Description	The radiation length of the material. This keyword is only necessary if the material consists of only one material.
Example	`Silicon.RadiationLength 9.35`

The directory $(MEGALIB)/resource/geometries/materials contains a general file "Materials.geo", which includes all used materials for the MEGA geometries. The same directory contains also the absorption probability files. If you add a new material to Materials.geo, you also have to add the absorption probabilities to this directory. This is done in the following way:

1. Change to the GMega source directory: cd $(GMEGA)/src
2. Launch GeoMega, load the geometry and create the Geant files ugeom.f and detinit.f (Geomega: *Menu: Analysis → Create Geant3 files*)
3. Switch to the main GMega directory and compile: cd $(GMEGA); make
4. Switch to the directory, where you want to have your absorption files, e.g. $(MEGALIB)/resource/geometries/materials
5. Run the absorptions program: absorptions
6. Now all absorption files are generated and you are done!

6.2.3 The object Volume

Please take notice of all distances except radii being half distances!

Keyword	**Volume** <text>
Description	Declares a volume with name <text>
Example	`Volume Wafer`

Keyword	<Volume>.**Shape** <BRIK> <half x-size> <half y-size> <half z-size>
Keyword	<Volume>.**Shape** <SPHE> <radius min> <radius max> <theta min> <theta max> <phi min> <phi max>
Keyword	<Volume>.**Shape** <TUBS> <radius min> <radius max> <half height> <phi min> <phi max>
Keyword	<Volume>.**Shape** <CONE> <half height> <bottom radius min> <bottom radius max> <top radius min> <top radius max>
Keyword	<Volume>.**Shape** <TRD1> <half distance x1> <half distance x2> <half distance y> <half distance z>
Keyword	<Volume>.**Shape** <TRD2> <half distance x1> <half distance x2> <half distance y1> <half distance y2> <half distance z>
Keyword	<Volume>.**Shape** <TRAP> <half distance in z> <theta> <phi> <half height trapezium bottom> <half bottom length trapezium bottom> <half top length trapezium bottom> <alpha trapezium bottom> <half height trapezium top> <half bottom length trapezium top> <half top length trapezium top> <alpha trapezium top>
Keyword	<Volume>.**Shape** <GTRA> <half distance in z> <theta> <phi> <twist> <half height bottom trapezium bottom> <half bottom length trapezium bottom> <half top length trapezium bottom> <alpha trapezium bottom> <half height trapezium top> <half bottom length trapezium top> <half top length trapezium top> <alpha trapezium top>

Keyword <Volume>.**Shape** <PCON> < the azimuthal angle phi at which the volume begins (angles are counted counterclockwise)> <opening angle of the volume> <number of sections, number should be at least 2, the following three arguments are repeated accordingly> <height> <minimum radius> <maximum radius>

Keyword <Volume>.**Shape** <PGON> < the azimuthal angle phi at which the volume begins (angles are counted counterclockwise)> <opening angle of the volume> <number of sides of the cross section between the given phi limits> <number of sections, number should be at least 2, the following three arguments are repeated accordingly> <height> <minimum radius> <maximum radius>

Description Describes the shape of the object.
BRIK: A box
SPHE: A sphere
TUBS: A cylinder
CONE: A cone
PCON: A Polycone
PGON: A Polygon
TRD1, TRD2, TRAP, GTRA: trapezoids (try to avoid TRAP and GTRA!)
Detailed descriptions of the objects can be found in the Geant3, Geant4 or ROOT manual (ROOT comes with pictures!). The argument list corresponds to that of ROOT. If you want new volumes to be added, let me know, but make sure such a volume exists in Geant3, Geant4 and ROOT and provide an equation for its volume! One limitation for the current Geant4 implementation is that TRAP and GTRA are not allowed to be triangles (i.e. non of the values height, bottom and top length is allowed to be zero). So if you need triangles make sure, they contain very small values.

Example
```
Wafer.Shape BRIK 0.0 0.0 30.0
GeLeft.Shape TRAP 3.5 0 0 0.86602540378 3 0.0000000000001 0
0.86602540378 3 0.0000000000001 0
```

Keyword <Volume>.**Material** <Material name>
Description Material of the volume. It has to be defined somewhere else in the setup file.
Example
```
Wafer.Material Silicon
```

Keyword <Volume>.**Visibility** <0 or 1>
Description If the value is zero, then the volume is not visible, if it is 1, the volume is visible in the geomega viewer.
Example
```
Wafer.Visibility 1
```

Keyword <Volume>.**Color** <int>
Description The color of the volume when display in the viewer. The numbers correspond to the ROOT color IDs.
Example
```
Wafer.Color 5
```

Keyword <Volume>.**Mother** <Name of volume>
Description In this mother volume the volume will be placed. Attention: the volume must be fully contained in her mother and the mother must be defined somewhere else in the setup file. For the world volume
Example
```
Wafer.Mother Tracker
```

Keyword <Volume>.**Position** <x> <y> <z>
Description Position of the volume within its mothers coordinate system. Position is the center of the volume. Attention the volume has to be fully contained in its mother volume!
Example
```
Wafer.Position 0.0 0.0 30.0
```

Keyword	<Volume>.**Rotation** <x> <y> <z>
Description	Counterclockwise rotation of the volume around x-, y-, and z-axis in the mother volume. Attention the volume has to be still fully contained in its mother volume!
Example	`Wafer.Rotation 0.0 0.0 90.0`

Keyword	<Volume>.**Rotation** <theta1> <phi1> <theta2> <phi2> <theta3> <phi3>
Description	Rotation according to the Geant3 convention: <theta1> is the polar angle of the x-prim axis in the main reference system (MRS), <theta2> and <theta3> have the same meaning for the y-prim and z-prim axis. <phi1> is the azimuthal angle of the x-prim in the MRS and <phi2> and <phi3> have the same meaning for y-prim and z-prim. Attention the volume has to be still fully contained in its mother volume!
Example	`Wafer.Rotation 90 90 0 0 90 0`

Keyword	<Volume>.**Scale** <double>
Description	Scale (shrink or enlarge) a volume and all sub volumes. Useful if one has a large volume tree and wants to modify it's size. This keyword can not be applied to Copies/Clones.
Example	`SpaceCraftBody.Scale 0.5`

6.2.4 Detector objects

There exist 7 different detector types: A 2D strip detector like the MEGA Silicon wafers (2DStrip), a 3D strip detector (3DStrip) like the NCT Germanium detectors, directional 3D strip detectors, where some information of the electron direction is retained, a Drift Chamber detector including capabilities for light sensing – needed for liquid Xe and gas microwell detectors, a calorimeter like in MEGA and a one-volume-type detector, which can be used as MEGA ACS as well as SPI Germanium detector, or any other thumb detector, which can only measure energy information, and finally a 3D voxel detector.

6.2.4.1 The detector keywords .

Keyword	**Strip2D** <detector name>
Description	Declares a 2d strip detector with name <detector name>. The strips are always oriented in x and y direction. If you want another orientation, simple rotate the detector volume.
Example	`Strip2D bachus`

Keyword	**Strip3D** <detector name>
Description	Declares a 3D strip detector with name <detector name>. It inherits all capabilities from Strip2D. In addition it has a depth resolution, which is always the z direction.
Example	`Strip3D nct1`

Keyword	**Strip3DDirectional** <detector name>
Description	Declares a 3D strip detector with name <detector name>. It inherits all capabilities from Strip3D. In addition it has a directional resolution: It can detect all directions of electrons originating from Compton interaction. Due to limitations in the simulation, it is currently not possible to detect the direction of electrons simply passing through the silicon layer. If you want to use this detector, you need full "IA" information from the simulation. Thus if you use mggpod, make sure to use INIT2 and ACT2 options.
Example	`Strip3DDirectional SiWafer`

Keyword	**DriftChamber** <detector name>

Description Declares a drift chamber detector with name <detector name>. It inherits all capabilities from Strip3D. In addition, specific information like the (optical) light speed in this material ("LightSpeed"), the light sensitive detector side ("LightDetectorPosition") and its energy resolution ("LightEnergyResolution") as well as the electron drift parameters ("DriftConstant", "EnergyPerElectron") can be set.
Example `DriftChamber Chamber`

Keyword **Calorimeter** <detector name>
Description Declares a calorimeter with name <detector name>
Example `Calorimeter Fortuna`

Keyword **Scintillator** <detector name>
Description Declares a large, one channel and non position sensitive scintillation detector with name <detector name>. The special feature of this volume is, that it does not need to be box-like and that it can consists of multiple volumes. A detector of this type can also be used to model the SPI-Germanium detectors. Hits in this detector are always centered.
Example `Scintillator SPI12`

Keyword **Voxel3D** <detector name>
Description Declares a box-shaped volume consisting of voxels in all three dimensions. This detector has no known real-world counterpart, as it has no passive material (electronics, connectors, etc.) between the voxels. Its primary function is currently simulation diagnostics. Hits in this detector are always centered.
Example `Scintillator SPI12`

6.2.4.2 Common keywords .

Keyword <Detector>.**SensitiveVolume** <name of volume>
Description Volume, in which positions and energies of interactions are measured. Typical examples are one CsI-crystal or one single Si-wafer of the MEGA prototype.
Example `MEGACal.SensitiveVolume CsICrystal`

Keyword <Detector>.**DetectorVolume** <name of volume>
Description A larger volume which contains several evenly spaced sensitive volume. The position of the sensitive volumes are specified in the "Structural" parameters. They must be identical with the positions given in the volume description! Typical examples are the MEGA calorimeter, which consists of 120 CsI crystal or one layer of the MEGA tracker, which consists of 9 Si-wafers. The sensitive volume is either the same as the detector volume or entirely and **unrotated** contained in the detector volume!
Example `MEGACal.DetectorVolume CsIDetector`

Keyword <Detector>.**StructuralPitch** <x> <y> <z>
Description Spacing between the sensitive volumes (distance between the end of last sensitive volume to the start of the next sensitive volume). This is not the pitch between the individual strips of a strip detector! If you have a tracker then also add the distance between the layers as z-component!
Example `MEGACal.StructuralPitch 0.07 0.07 0.0`

Keyword <Detector>.**StructuralOffset** <x> <y> <z>
Description Distance between the edge of the detector volume to the beginning of the first sensitive volume. Calculated from negative to positive axis!
Example `MEGACal.StructuralOffset 0.235 0.235 0.185`

Keyword	<Detector>.**NoiseThreshold** <double>
Description	All hits in one voxel of the detector which are below this energy (in keV) are assumes not to be measured.
Example	`MEGACal.NoiseThreshold 50`

Keyword	<Detector>.**TriggerThreshold** <double>
Description	A hit need to deposit at least this energy (in keV) to raise a trigger signal.
Explanation	The difference between noise and trigger threshold: Each strip has a certain amount of electronics noise. So when one reads it out, one gets a signal in ADC counts, which only reflects the noise of the electronics. So normally one only uses hits which are well above the noise, e.g. 5 sigma above it. Trigger threshold is something different. A hit needs to produce a certain voltage (i.e. has to deposit a certain energy) to initiate the read-out of the detector.
Example	`MEGACal.TriggerThreshold 100`

Keyword	<Detector>.**EnergyResolutionAt** <energy> <measurements error>
Description	Energy resolution at the given energy in keV (one sigma)
Example	`MEGACal.EnergyResolution 662 60`

Keyword	<Detector>.**FailureRate** <double>
Description	Random failure rate – values are between 0 (no failures) and 1 (complete loss of detector)
Example	`MEGACal.FailureRate 0.1`

6.2.4.3 Keywords specific to calorimeters .

Keyword	<Detector>.**DepthResolution** <energy in keV> <noise in cm (one sigma)>
Description	Give the depth resolution at the given energy in cm (one sigma). If this keyword is not given, then no depth resolution is assumed, i.e. it behaves like a Strip2D detector.
Example	`MEGACal.DepthResolution 500 1.2`

6.2.4.4 Keywords specific to strip detectors (Strip2D, Strip3D, Voxel3D, DriftChamber) .

Keyword	<Detector>.**Offset** <x> <y> <z (Voxel3D only)>
Description	Distance between the edge of the sensitive detector to the beginning of the first strip
Example	`SMegaSD.Offset 0.142 0.142`

Keyword	<Detector>.**StripNumber** <x> <y> <z (Voxel3D only)>
Description	Number of strips
Example	`SMegaSD.StripNumber 128 128`

Keyword	<Detector>.**GuardringTriggerThreshold** <double>
Description	A hit in the gurad ring needs to deposit at least this energy (in keV) to raise a trigger (veto) signal.
Example	`SMegaSD.GuardringTriggerThreshold 100`

Keyword	<Detector>.**GuardringEnergyResolutionAt** <energy> <measurements error>
Description	Energy resolution at the given energy in keV (one sigma) for the guard ring
Example	`SMegaSD.EnergyResolution 100 10`

For 3D strip detectors (with and without electron direction resolution) and DriftChambers there is another special keyword:

Keyword	<Detector>.**DepthResolution** <Energy in keV> <Resolution in cm (one sigma)>
Description	Set the one sigma depth resolution. If this keyword is not given, then no depth resolution is assumed.
Example	`GeStrip.DepthResolution 200 0.2`

Keyword	<Detector>.**DepthResolutionThreshold** <Energy in keV>
Description	Below this threshold no depth resolution can be measured, i.e. the center (z-axis) of the detector is used as z-position.
Example	`GeStrip.DepthResolutionThreshold 25`

For 3D strip detectors with directional resolution exists another special keyword:

Keyword	<Detector>.**DirectionalResolution** <Energy in keV> <Resolution in degree (one sigma)>
Description	Set the one sigma directional resolution.
Example	`SiStrip.DirectionalResolution 200 30`

DriftChambers have additional keywords

Keyword	<Detector>.**LightSpeed** <light speed in cm/s>
Description	Light speed of the scintillation light in the drift chamber
Example	`Chamber.LightSpeed 18E+9`

Keyword	<Detector>.**LightDetectorPosition** <0: none; 1: +x; -1:-x; 2: +y; -2 -y; 3: +z; -3:-z>
Description	Represents the side of the detector which is light sensitive (i.e. equipped with PMTs or diodes). If this value is zero, then no light detector is assumed.
Example	`Chamber.LightDetectorPosition 3`

Keyword	<Detector>.**DriftConstant** < constant in cm >
Description	One sigma of the opening cone of the drift in the E-field ($c \times \sqrt{Distance\ to\ wires}$) . If this is zero, a simple and fast projection is used.
Example	`Chamber.DriftConstant 0.01`

Keyword	<Detector>.**EnergyPerElectron** <Energy in keV>
Description	Energy of one drifting electron
Example	`Chamber.DriftConstant 0.01`

Keyword	<Detector>.**LightEnergyResolutionAt** <Energy> <Resolution in keV (one sigma)>
Description	One sigma energy resolution of the detected light.
Example	`Chamber.DriftConstant 0.01`

6.2.4.5 Keywords specific to ACS-like detectors .

The special feature of this detector type is that it may contain several sensitive volumes of different shapes.

Keyword	<Detector>.**HitPosition** <Volume name 1> <Volume name 2> <x> <y> <z>
Description	A hit in volume <Volume name 1> is moved by x, y, z into <Volume name 2>. All sensitive volumes of the detector have to be covered!
Example	`GeDet.HitPosition GeCentral GeCentral 0 0 0`

```
GeDet.HitPosition GeLeft GeCentral 0 -2.59807621135 0
GeDet.HitPosition GeRight GeCentral 0 -2.59807621135 0
```

6.2.5 Triggers

The current implementation of triggers is not completely finished and not obvious throughout the package: It has only been tested for MEGA type triggers (D1+D2) and the simulation does not necessarily take into account all options: While GMega uses part of this information (it only translates the info to "TriggerByChannel" and "DetectorType", which is the more open trigger and GMega ignores vetoes and trigger thresholds), Cosima has in its setup file an own trigger condition. The approach is the following: First define if this trigger is a veto or not. Then define if the triggering happens by channel or by detector. "TriggerByChannel" means that the channels are counted for the trigger, "TriggerByDetector" means that the detectors are counted (irrelevant of how many channels in this detector have trigger - if at least one is above the trigger threshold). Then define the triggering detectors (by type or name) and the number of hits they must accumulate. Pay attention, you can only have one type of detector keyword per trigger, i.e. one trigger is allowed to contain only "DetectorType" keywords or only "Detector" keywords or only "GuardringDetectorType" keywords or only "GuardringDetector" keywords!

Keyword	**Trigger** <text>
Description	Declares a trigger condition with name <text>
Example	`Trigger D1D2`

Keyword	<Trigger>.**Veto** <true/false>
Description	Determines if this is a real trigger (acceptance) or a veto trigger (rejection). If this option is not given then veto is false!
Example	`ACSVeto.Veto true`

Keyword	<Trigger>.**TriggerByChannel** <true/false>
Description	Determines that the hits are accumulated channel wise. This is the default.
Example	`D1D2.TriggerByChannel true`

Keyword	<Trigger>.**TriggerByDetector** <true/false>
Description	Determines that the hits are accumulated detector wise. TriggerByChannel is the default
Example	`D1D2.TriggerByChannel true`

Keyword	<Trigger>.**DetectorType** <Detector type name> <Number of hits>
Description	<Number of hits> are necessary in detector type <Detector type name> to raise a trigger. If this keyword occurs multiple times all conditions have to be fulfilled. The <Detector type name> follows the MEGAlib convention: *Strip2D* is a 2D strip detector, *Calorimeter* is a MEGA calorimeter, *Strip3D* is a 3D strip detector, *Scintillator* is a scintillator/ACS type detector and *DriftChamber* is of course the drift chamber.
Example	`D1D2.DetectorType Strip2D 1` `D1D2.DetectorType Calorimeter 1`

Keyword	<Trigger>.**Detector** <Detector name> <Number of hits>
Description	<Number of hits> are necessary in <Detector name> to raise a trigger. If this keyword occurs multiple times all conditions have to be fulfilled.

Keyword	<Trigger>.**GuardringDetectorType** <Detector type name> <Number of hits>

Description	<Number of hits> are necessary in the guard ring of detector type <Detector type name> to raise a trigger. If this keyword occurs multiple times all conditions have to be fulfilled. The <Detector type name> follows the MEGAlib convention: *Strip2D* is a 2D strip detector, *Calorimeter* is a MEGA calorimeter, *Strip3D* is a 3D strip detector, *Scintillator* is a scintillator/ACS type detector and *DriftChamber* is of course the drift chamber.
Example	`D3.GuardringDetectorType Strip3D 1`

Keyword	<Trigger>.**GuardringDetector** <Detector name> <Number of hits>
Description	<Number of hits> are necessary in the guard ring <Detector name> to raise a trigger. If this keyword occurs multiple times all conditions have to be fulfilled.
Example	`D3.GuardringDetector MyStrip3D 1`

Here are more examples:
The MEGA prototype has an electron tracker and a calorimeter. Thus, a reasonable trigger condition would require at least two layers of the tracker and one calorimeter triggering. In addition any events with hits in the veto dome should be rejected. Thus we define two trigger conditions:

```
Trigger Main
Main.Veto false
Main.TriggerByDetector true
Main.Detector MyD1 2
Main.Detector MyD2 1

Trigger AntiCoincidence
AntiCoincidence.Veto true
AntiCoincidence.TriggerByDetector true
AntiCoincidence.Detector MyAnticoidence 1
```

A thick Germanium detector might require at least three hits for Triple Compton coincidence. Since the detector is thick, those hits do not need to be in different detectors but only different channels need to trigger. In addition, events which deposit energy in the guard rings of the detector are going to be rejected:

```
Trigger Main
Main.Veto false
Main.TriggerByChannel true
Main.DetectorType Strip3D 3

Trigger Guardring
Guardring.Veto true
Guardring.TriggerByDetector true
Guardring.GuardringDetectorType Strip3D 1
```

The triggering is performed during reading the events from the *.sim file into revan or sivan. No triggers are tested for the *.evta files, since those events represent detector data. A trigger is raised, when the deposited energy in the given channel is above the trigger threshold of the detector or guard ring.

6.2.6 Additional features

6.2.6.1 Templates and the keyword Copy .

In a usual detector geometry some volumes will appear several times. To avoid any copy and paste, the keyword Copy has been introduced, which copies the characteristics of one volume to another. Normally the base volume is defined as template, i.e. it does not have a mother and it is not positioned.

```
Volume GeWafer
GeWafer.Material Germanium
GeWafer.Visibility 1
GeWafer.Color 6
GeWafer.Shape BRIK 4.0 4.0 1.0

// Arrange the Ge-detectors
GeWafer.Copy GeE1N001
GeE1N001.Position 0.0 0.0 3.75
GeE1N001.Mother WorldVolume

GeWafer.Copy GeE1N002
GeE1N002.Position 0.0 0.0 1.25
GeE1N002.Mother WorldVolume
```

Up to now, the keyword Copy is only implemented for volumes!

Keyword	<Volume>.**Copy** <existing volume>
Description	Create the new volume <existing volume> and copy all characteristics of <Volume> to the new volume
Example	`SiLayer.Copy bachus`

6.2.6.2 The keyword Virtual .

One major restriction of the geomega format is that no overlapping volumes are allowed. But sometimes it is useful to have a larger volume, in which several other volumes are grouped, to use as template. In the case this larger volume has no real meaning other than being a container for other volumes, it can be declared as virtual. Those virtual volumes are allowed to overlap with other volumes – as long as their content does not overlap with any other volume! During the creation of the geometry, virtual volumes are removed from the volume tree!

Keyword	<Detector>.**Virtual** <bool>
Description	Declare a volume as virtual
Example	`TrackerContainer.Virtual true`

6.2.6.3 The keyword Constant .

A very experimental feature is that one can define constants in the setup file:

Keyword	<Constant> <string: Replace this> <string: with this>
Description	Replace some text with another text
Restrictions	It is not checked if the string is a token, it is even replaced if it is part of another string. Don't use any keywords!
Example	`Constant Size 2.0`
Example	`Wafer.Shape BRIK Size Size 0.025`

6.2.6.4 The maths environment .

Another very experimental feature is that one can do very basic mathematical calculations in the setup file. The signs { and } start and end the maths environment. Everything within those brackets is considered as one token:

```
Constant Size 2.0
Wafer.Shape BRIK {1.2*Size} {1.2*Size} {0.5*(log(Size)-1)}
```

The maths environment relys on the ROOT interpreter. So whatever manipulation ROOT knows can be done in the maths environment.

6.2.7 Closing remarks

Before you start coding a geometry, take a look at the examples in the directory $(MEGAlib)/resources/geometries. Try to write your code as close as possible to the examples, in order to avoid any trouble.
It is very time-consuming to protect against each possible input error in an open file format like this. So do not expect to get a warning/error message for each time you make a mistake – sometimes the program will warn you with an error message, sometimes it will simply accept the error and produce false results and sometimes it will simply crash. Whenever you come across such a problem, let me know and I will implement a protection.
Something which is very difficult to implement is a check for overlapping volumes – so this is not done at the moment. Therfore it is the users responsibility to make sure no volumes overlap.
In addition, since most of the input parameters need to be positive and almost no sanity checks for positivity are implemented, it is the users responsibility to ensure positive values where reasonable.

6.2.8 Practices for writing a good geometry

- Use meaningful descriptions, e.g. WorldVolume instead of VAC0
- Don't make your geometry flat! The more daughters a volume has, the more volumes have to be searched when a particle moves from one volume to the next! It i significantly better to have a steep geometry: The world volumes contains daughters, which contain daughter volumes, which contain daughter volumes, etc.
- Use multiple files representing different objects: If you have distinguishable objects such as individual detectors with all their surround electronics and mounting, describe them in an individual file! Simply "Include" this file in the file where its mother volume is described. This makes your geometry much more clearly laid out.

6.3 The TRA format

6.3.1 Common key words

Start event
SE

Event type
ET <Type $\in \{PA, CO\}$ >

Event ID (same as in SIM)
ID <int>

Time in seconds Linux time
TI <Time in sec>

Time walk between D1 and D2 trigger in ns - only for real MEGA data
TW <Time in ns>

x- and z-axis of orientation of detector (spherical coordinates)
RX <x> <y> <z> (Default: 0 1 0)
RZ <x> <y> <z> (Default: -1 0 0)

Orientation of instrument in galactic coordinates
GX <Longitude in degree> <Latitude in degree>
GZ <Longitude in degree> <Latitude in degree>

6.3.2 Compton events

Energies in keV:
CE <Energy of scattered gamma-ray>
<Energy error of scattered gamma-ray>
<Energy of recoil electron>
<Energy error of recoil electron>

Positions and directions of the first interaction
CD <x position of first interaction in cm>
<y position of first interaction in cm>
<z position of first interaction in cm>
<x position error of first interaction in cm>
<y position error of first interaction in cm>
<z position error of first interaction in cm>
<x position of second interaction in cm>
<y position of second interaction in cm>
<z position of second interaction in cm>
<x position error of second interaction in cm>
<y position error of second interaction in cm>
<z position error of second interaction in cm>
<Electron direction x>
<Electron direction y>
<Electron direction z>
<Error of electron direction x>
<Error of electron direction y>
<Error of electron direction z>

Length of Compton sequence in "detected interactions"
SQ <Length (1, 2, 3, etc.)>

First and second Compton quality factor (aka. "Compton test statistics")
CT <Probability best sequence> <Probability second best sequence>

Length of first track
TL <Length (1, 2, 3, etc.)>

Deposited energy of first track in its first layer
TE <Energy in keV>

Shortest distance in the sequence (not first distance)
LA <Distance in cm>

Time-of-flight between first and second interaction
TF <Time-of-flight in s> <Uncertainty of time-of-flight in s>

COMPTEL PSD value
PD <PSD a.u.> <Uncertainty PSD a.u.>

6.3.3 Pair events

Position of the gamma conversion
PC <x in cm> <y in cm> <z in cm>

Energy and initial direction of the first generated particle
PE <Energy in keV> <Energy error in keV> <Direction x> <Direction y> <Direction z>

Energy and initial direction of the second generated particle
PP <Energy in keV> <Energy error in keV> <Direction x> <Direction y> <Direction z>

Energy deposit in the initial layer
PI <Energy in keV>

7 ConvertMGGPOD

7.1 Command line options

Mandatory:	
-f <filename>	Load this fits file (*.fits) (this command can appear several times!)
-g <filename>	Load this geometry file
Optional:	
-o <filename>	Name of the output file. Per default *.sim is appended to the base name of the fits file.
-m	Sets the maximum number of steps, which the fits events may have per detector type before the discretization (default: 5000)
-s <int>	Version number of the generated *.sim file. Default is 3, for Bayesian event reconstruction you need 22.
-i	Ignore hits which consist of only one hit
–use-nohits-events	Writes output also for those events, which do not have a hit
–ignore-veto	To reduce the file size, events which produced a veto are per default thrown away. Use this flag ignore all vetos. (if this flag is NOT set, the veto is applied at this step, using the energy threshold for the anticoincidence detector(s) given in the geometry file. NO noising is applied to the veto — so the result is a step-function veto threshold at the nominal threshold energy! Later, revan will apply noising and another veto decision to the remaining events!)
–no-discretization	Do not do the discretization of hits into voxels
–no-gui	Avoids the progress bar screen
-d	Dump the content of the fits file. Special option, which ignores everything but -f & -g!

8 GeoMega - Geometry for Mega

8.1 Command line options

Geomega is a GUI based application. Thus, you do not need to give command line options. In fact only the most frequently used options are implemented as special tasks.

Optional:	
-g *or* –geometry <filename>	Load this geometry file (*.geo.setup)
-d	Activates debug mode
-s *or* –startvolume <name>	Use this volume as world volume. If this volume exists as several copies, use one of the Copy names!
Special tasks:	
–create-mggpod <filename suffix>	Create mggpod files with this file name suffix

–create-mggpod-default	Create mggpod files with the default file names (setup.geo, media.med, materials.mat)

9 Revan - Real event analyzer

9.1 Command line options

Revan is a GUI based application. Thus, you do not need to give command line options. In fact only the most frequently used options are implemented as special tasks.

Optional:	
-f <filename>	Load this sim or evta file (*.sim, *.evta)
-g <filename>	Load this geometry file (*.geo.setup)
-d	Activates debug mode
–no-gui	Avoids the progress bar screen
Special tasks:	
-a <filename>	Load this configuration file and start the reconstruction of the events. If -f and -g are given, then those override the values of the configuration file.

9.2 Graphical user interface

Menu File:	
Open	Open a simulation (*.sim) or events (*.evta) file
Load Geometry	Load the geometry, which belongs to the simulation or events file.
Load Configuration	Load a configuration file, which contains all GUI information.
Save Configuration	Save a configuration file, which contains all GUI information.
Exit	
<list>	List of previously used revan files
Menu Reconstruction:	
Start Event Reconstruction	Starts the event reconstruction
Selection of algorithm	Selects different reconstruction algorithms – the following four menus will have different contents for different algorithms
Clustering options	**No clustering:** No options **Clustering by distance:** Enter the maximum distance between two interactions in cm up to which to hits are combined to a cluster *Use center of the cluster (instead of each sub hit individually) as reference point for distance calculations* : In default mode, if one has already a cluster to which an additional hit should be added, then all sub-hits are checked if the new hits is within the distance. If the option is checked, then new hits have to be within a minimum distance radius around the center of gravity/energy of the cluster to be added to the cluster. **Clustering of adjacent voxels:** Select the numbers of neighbors (4 closest neighbors or 8 clostest neighbors) of the adjacent voxels. The clusterization can be aborted if the z-distance is larger than a certain sigma of the position resolution. Ignore this if the sigma is negative.
Tracking options	**No tracking:** No options **General options for all tracking algorithms:**

Check if the algorithm should check for pair events, MIPs or Compton electron tracks.
Maximum layer jump for Comptons offers to set the maximum allowed distance of two hits (in layer distances). One means the hits have to be in consecutive layers, two means that there is at maximum one layer between the hits.
Maximum number of different track sequences to keep This means that not only the best, but the two, three, four, five, best track sequences are transferred to CSR. For MEGA this did not turn out to improve the performance (but worsens it), but is intended for thick Silicon trackers for handling two layer interactions.
Reject purely ambiguous sequences Do not try to analyze tracks which have not any straight beginning, but play ping-pong between the layers, or which curl. Those tracks are more likely to be wrongly reconstructed, but at low energies there are a lot of them...
Bayesian electron tracking
Give the *file containing the tracking data* . This has to be the file ending with *.goodbad.rsp

Sequencing options

No Compton tracking
No options
Compton sequence reconstruction without energy recovery
Use two site events without track means to use events which consist of nothing but two hits
Only use events which start in 2D strip detectors rejects events which start in anything else but a 2D strip detector
Reject events which are limited to one detector type rejects events which do not interact at least in two different detector types like 2D strip detector and 3D strip detector or 2D strip detector and calorimeter
Assume undecided events start in 2D strip detectors (tracker) assumes for all 2 site events, whose direction of motion can not be determined, that they start in a silicon strip detector
Always assume events starts in the 2D strip detectors (tracker) always forced 2 site events to start in the tracker,
Allowed range of Compton quality factor rejects events whose final quality factor is not within the limits
Maximum number of interactions: Since the complexity of the algorithm goes with the factorial of the number of interactions, we have to cut it off here.
All Comptons have to originate from objects in this file checks if the Compton cone intersects volumes in the given geometry file. Only those events are kept.
Compton sequence reconstruction with energy recovery
tbd.
Bayesian Compton tracking
Give the *file containing the BCT data* , i.e. the files containing the Bayesian matrices. You only need to give only file, the others are assumed to reside on the same directory. This has to be the file ending with *.goodbad.rsp

Maximum number of interactions: Since the complexity of the algorithm goes with the factorial of the number of interactions, we have to cut it off here.

Compton sequence reconstruction with time of flight
tbd.

Decay option

No decay detection
No options
Decay detection for Comptons after Compton tracking
File containing decay lines wants the file, which contains the individual energies of the decays, for which the algorithm should search.

General event selections	Select events by their total deposited energy, their minimum lever arm between any two Compton interactions or event ID (negative numbers means no ID limit)
Menu Applications:	
Initial energy distribution	Shows the energy distribution before the event reconstruction
Energy distribution before/after	Compares the energy distribution before and after the event reconstruction
Energy per cluster per detector type	For each detector type show a histogram of how many energy per cluster is deposited
Spacial hit distribution	Shows a 3D hit distribution
Number of clusters	Dumps a cluster distribution

10 Mimrec - Mega image reconstruction

10.1 Command line options

Optional:	
-f <filename>	Load this sim or evta file (*.sim, *.evta)
-g <filename>	Load this geometry file (*.geo.setup)
-c <filename>	Load a configuration file (*.cfg)
-d	Activates debug mode
Special tasks:	
-i	Create an image.
-s	Create a spectrum
-a	Create an ARM
-e	Dump the event selections
Additional options for the special tasks	
-n or –no-gui	Do not use a graphical user interface
-k or –keep-alive	Do not quit after executing a batch run, if we do have a gui

10.2 Graphical user interface

Menu File:	
Open	Open a simulation (*.sim) or events (*.evta) file
Load Geometry	Load the geometry, which belongs to the simulation or events file.
Load Configuration	Load a configuration file, which contains all GUI information.
Save Configuration	Save a configuration file, which contains all GUI information.
Exit	
<list>	List of previously used revan files
Menu Backprojection:	
Coordinate system	Choose between the three different coordinate systems: Spherical, Cartesian 2D and Cartesian 3D
Image dimension	Choose the range of the different axes and the number of bins (Do not modify the image axis vector!)
Backprojection algorithm	Choose between the backprojection to points and areas. The shape of the area is controlled by later GUIs
Response selection	Only Universal adaptable Compton response is currently working

Response parameters	Give the length (longitudinal) of the Compton cone (for tracks only) and the width (transversal) of the Compton cone as well as the width of the pair response - all values are 1 sigma. If you want to use absorptions probabilities be prepared to wait for the result... The fit parameter file is currently not working.
Sensitivity	Not working
Event selections	Choose from a wide range of different event selections. For somebody with a basic knowledge of Compton scattering/Compton telescopes most of them should be obvious. The only sophisticated one is the earth horizon cut: You have the choice between: (1) No cut (2) A cut which rejects all events which intersect the horizon - this cut does not differentiate between tracked and not tracked events. Tracked events are assumed to be a full circle. (3) A probability based cut. It is calculated how likely it is that the event came from earth. If this probability exceeds the given value (GUI: maximum earth probability), then the event is rejected. For not tracked events this probability is determined by the portion of the cone which came from below the horizon. For tracked events this is a little bit more complicated and needs a response file, which has to be given via the GUI: The response file is basically a 2D-distribution, Energy versus SPD, and allows to determine the portion of the arc, which is below the horizon. If no response file is given in probability mode, we default to untracked mode! The location of the horizon is given by the last GUI element: "Angle between direction of earth center and horizon". In the GUI, the earth is currently fixed to be at the nadir of the detector (although everything else can already handle any earth position - theoretically). If you enter 0 deg in the GUI then you basically have no earth horizon, if you enter 90 deg then the horizon is your lower hemisphere.
Memory management	You normally want only use 1 byte for the response matrix
Menu Likelihood:	
Algorithm	OS-EM is broken, so use Classic EM. As stop criteria you can use the number of iterations (recommended) or a minimum allowed likelihood change.
Penalty	Broken
Menu Reconstruction:	
Start image reconstruction	Nomen est omen
Stop image reconstruction	Nomen est omen
Menu Response:	If somebody doesn't understand the menu names, he is not worth working with mimrec! To all those function here the event selections set in *Menu: Backprojection → Event selection* apply. Some points (angular resolutions, energy spectra) open an additional window where the position of the original source can be set and a window (radius) around this position. If that window is negative, it is ignored.

11 SensitivityOptimizer

Calculates the point source sensitivity based on a source file and several background files in a multi-dimensional data space. The result (point of best sensitivity in the data space) dumped to a log file.

11.1 Command line options

General options:

-n <name>	name (suffix) for the output files

-s <file> <sim photons> <sim area>	source file, including the number of simulated photons and their start area
-b <file> <time>	background file name and measurement time
-c <file>	Mimrec Configuration File: This has to be an file, with no event restrictions in Compton Sequence Length, Earth horizon cut, Compton scatter angle, Compton quality factor, ARM and Energy!
-w <min> <max>	use a larger energy window for collection of background events (in keV)
–csl <min low edge> <min high edge> <max>	Compton sequence length: the minimum varies between its low and high edge, the maximum is fixed to max. "2 2 2" means there is one bin representing two-site events, "2 2 7" means there is one bin representing hits with a length between 2 and 7, "2 7 10" we have six bins, representing the lengths 2 to 10, 3 to 10, 4 to 10, 5 to 10, 6 to 10 and 7 to 10
–ehc <min> <max> <steps>	Earth horizon cut in deg (0..min..max). Null means no cut
–phi <min> <max> <steps>	Compton scatter angle in deg (0..min..max)
–cqf <min> <max> <steps>	Compton quality factor (0..min..max)
–pos <theta> <phi>	Position of the source
–arm <min> <max> <steps>	ARM in deg (Radius: 0..min..max)
–spd <min> <max> <steps>	SPD in deg (Radius: 0..min..max) - not necessary for not tracked events!
–egy <mean> <min> <max> <steps>	Energy in keV (Radius: 0..min..max)
-h	Print this help

11.2 Example

SensitivityOptimizer
-s MySims/Source847bl/Source847bl.tra 2000000 41547.56
-b MySims/AlbedoPhotons/AlbedoPhotons.tra 2759.52
-b MySims/CosmicPhotons/CosmicPhotons.small.tra 6441.8
-b MySims/CosmicProtons/CosmicProtons.prompt.tra 376.752
-b MySims/CosmicProtons/CosmicProtons.activation.tra 12000.0
-b MySims/TrappedProtons/TrappedProtons.activation.tra 16000.0
-c Open.cfg
–csl 3 3 7 –ehc 0 100 6 –phi 80 140 4 –cqf 0.4 1.0 7 –pos 0 0 –arm 1.0 2.0 6 –spd 10.0 30.0 5 –egy 847 16.0 16.0 1
-n 847blSens

12 How-to guides

12.1 Use Case 1

This example shows the data analysis path from simulation to imaging for two ^{88}Y sources located on top of the iglu of the MEGA prototype.

12.1.1 Generation of a geometry file for GMega

- Run geomega:
  user@earth:~/MEGAsoft/MEGAlib> `geomega`

- Load the geometry file: *Menu: File → Open* and select the geometry `Prototype.geo.setup` from the directory `<path to MEGAlib>/resource/geometries/simplifiedprototype`. Alternatively use the command line switch "geomega -f <file name>" to load the geometry.
- View the geometry: *Menu: Analysis → View geometry* . If you don't see the geometry, check the command line for errors and correct them!
- Generate the Geant3 geometry information with *Menu: Analysis → Create Geant3 files* . This generates the files `ugeom.f` and `detinit.f`. Here `ugeom.f` contains the geometry information and `detinit.f` the detector information.
- When the files are generated, copy them into the `src` directory of GMega:
  user@earth:~/MEGAsoft/MEGAlib> `cp ugeom.f <path to GMega>/src`
  user@earth:~/MEGAsoft/MEGAlib> `cp detinit.f <path to GMega>/src`
- Switch to the GMega directory and recompile:
  user@earth:~/MEGAsoft/GMega> `gmake`

12.1.2 Generation of a geometry file for MGeant/MGGPOD

- Switch to the directory where you need the MGeant/MGGPOD files
- Run geomega:
  user@earth:~/MEGAsoft/MEGAlib> `geomega`
- Load the geometry file: *Menu: File → Open* and select the geometry `Prototype.geo.setup` from the directory `<path to MEGAlib>/resource/geometries/simplifiedprototype`. Alternatively use the command line switch "geomega -f <file name>" to load the geometry.
- View the geometry: *Menu: Analysis → View geometry* . If you don't see the geometry, check the command line for errors and correct them!
- Generate the MGGPOD/MGeant geometry information with *Menu: Analysis → Create MGeant/MGGPOD files* . This generates the files `media.med`, `materials.mat`, and `setup.geo`.
- Now you are ready to run MGeant or MGGPOD

Cosima does not need any special generated geometry file: You only announce the name of the geometry in the *.source file.

12.1.3 Generation of a simulation file

12.1.3.1 Generation of a simulation file with GMega

- First make sure that the file Restart.dat does not exist. This file contains restart information in case the last run has been interrupted:
  user@earth:~/MEGAsoft/GMega> `rm -f Restart.dat`
- Then start GMega:
  user@earth:~/MEGAsoft/GMega> `GMega sgeo/UseCase1.src`
- ... and wait until the simulation has finished.

12.1.3.2 Generation of a simulation file with MGeant/MGGPOD

- Make sure you compile MGeant/MGGPOD with the ACT2/INIT2 extension (check `Imakefile`)
- The result of MGeant/MGGPOD is a fits file. This has to be converted into the MEGAlib format via the tool ConvertMGGPOD:
  user@earth:~/MEGAsoft/GMega> `ConvertMGGPOD -f UseCase1.fits -g $MEGALIB/resource/geometries/simplif`
- ... now you have a sim file which can be used with MEGAlib.

12.1.3.3 Generation of a simulation file with Cosima

- After you have successfully compiled Cosima, simply launch it with the appropriate source file:
  user@earth:~/MEGAsoft/Cosima> `Cosima source/UseCase1.source`
- ... and wait until the simulation has finished.

12.1.4 Analyzing the data

12.1.4.1 Analyzing the data with Sivan

- Start sivan:
  user@earth:~/MEGAsoft/MEGAlib> `sivan`
- Load the simulation file: *Menu: File → Open* . Switch to the directory GMega or Cosima and select the simulation file `UseCase1.sim` or `UseCase1.inc1.id1.sim`.
- The second step is to load the geometry belonging to the simulation file: *Menu: File → Load geometry* . In the directory `<path to GMega>/resource/geometries/simplifiedprototype` select the geometry `Prototype.geo.setup`.

  The last step is not necessary, if you have generated your simulation with Cosima, because then the path to the file is included in in the sim file.

  In case the geometry is wrong, the analysis will stop with a command similar to: `Cannot noise hit ...`
- Start the analysis with *Menu: Analysis → Analyze events and create *.tra file* This creates a *.tra file which contains the interaction information.

12.1.4.2 Analyzing the data with Revan

- Start revan:
  user@earth:~/MEGAsoft/MEGAlib> `revan`
- The second step is to load the geometry belonging to the simulation file: *Menu: File → Load geometry* . Switch to the directory `<path to MEGAlib>/resource/geometries/simplifiedprototype` and select the geometry `Prototype.geo.setup`.
- Choose all options from *Menu: Reconstruction → Selection of algorithm* and **below** which are reasonable. Pay attention that if you use the Bayesian algorithms you need to have generated Response files in advance. Since this is a difficult process, stay with the figure-of-merit algorithms right now.
- Start the analysis with *Menu: Reconstruction → Start event reconstruction* . This creates a *.tra file which contains the interaction information.

12.1.5 Point source imaging in the near-field

- Start mimrec:
  user@earth:~/MEGAsoft/MEGAlib> `mimrec`
- Load the *.tra file: *Menu: File → Open* . Switch to the directory GMega or Cosima and select the simulation file `UseCase1.tra` or `UseCase1.inc1.id1.tra`.
- Set the coordinate system to Cartesian 2D in *Menu: Backprojection → Coordinate system* .
- Set the correct zoom via *Menu: Backprojection → Image dimensions* . Set the following coordinates: $x_{min} = -5$, $x_{max} = +5$, $x_{bins} = 50$, $y_{min} = -5$, $y_{max} = +5$, $y_{bins} = 50$, $z_{min} = 40.9$, $z_{max} = 41.1$, $z_{bins} = 1$.

- The only reasonable back projection algorithm in *Menu: Backprojection → Backprojection algorithm* is Areas
- In *Menu: Backprojection → Response selection* use universal adaptable Compton response
- For the response fit parameters in *Menu: Backprojection → Response parameters* use Compton longitudinal = 60, Compton transversal = 1.5, Pair = 5.
- In *Menu: Backprojection → Event selections* make sure all event types are allowed except photo effect events (tracked and not tracked Comptons as well as Pair events). The good start point for the energy selection is 0/1200 keV. Make sure that the second energy range is set to on 0/0, that the time walk starts at negative values (e.g. -100), and that the energy range of the scattered gamma-ray starts at negative energies.
- In *Menu: Backprojection → Memory management* make sure that the response accuracy is set to one byte. 4 bytes are needed for elaborate schemes such as coded masks, where the response varies on small scales.
- If you want to look at the spectrum click *Menu: Response → Energy spectrum* . Ignore the position coordinates and set a negative radius. (If the radius is positive, then only those events are selected, whose Compton cone is within the radius distance from the given position.)
- If you want to get the image, then in *Menu: Likelihood → Algorithm* set the number of iterations to 10 and afterwards simply press the "Play" button.

12.2 Calculating the effective area of a telescope

The effective area of a telescope is the corresponding area of an ideal telescope which would detect all photons passing through its surface area (and measure their full energy). For Compton telescopes, the effective area is normally significantly smaller than the surface area.
The formula for calculating the effective area is:

$$A_{eff} = A_{start} \frac{N_{Photons\ within\ your\ energy\ and\ ARM\ selection}}{N_{Simulated\ events}} \quad (1)$$

The area from which the photons have been started can be obtained from the simulations.*If you have done the simulations with Cosima or GMega*, then look into your *.geo.setup. The start area is a disk whose radius is defined by the first parameter of the "SurroundingSphere". *If you have done the simulations with mggpod*, then you have to look into your fits file and locate the "BMPAR19" header information, which represents the area you look for. The number of simulated events is the input to your simulations.
To get the number of photons within your energy and ARM selection, follow these steps with mimrec:

- Do all your event selection in the mimrec event selector (Tracks yes or no, Length of Compton sequence, Compton quality factor, Range of Compton scatter angles, Earth horizon distance, etc.), with exception of the energy resolution. For the energy range, you take a window which is significantly larger than the peak width — but not to large — you want to have the peak width extend over at least 20 bins!
- Look at the energy spectra (the numbers for the GUI before the selection are: 0, 180, -1, 100). Do a fit: Right-click on the top of one of the green bins. You get a menu which is named "TH1D::EnergySpectrum". Select "Fitpanel". Select "gaus". Select with the bottom grey bar the fit window. Click fit. The screen output now tells you the sigma of your Gauss peak. Play a little around to find the best fit.
- Set a new energy window in the event selector (e.g. $\pm 1.4 \times$ 1-sigma energy resolution)
- Look at the ARM (the numbers for the GUI before the selection are: 0, 180, 5, 100 for on axis sources). The function is automatically fitted by 2 Lorentzians and one (asymmetric) Gaussian. If the fit is not ok, play with your bin size and ARM window ("Radius" parameter in the GUI). Make sure the FWHM of the peak is spread over at least 20 bins! The FWHM determined in the fit is written to the screen.

- Now do the fit again: As radius in the screen, which appears after clicking "ARM of scattered gamma-ray" set the FWHM of the ARM fit. Generate the ARM and look at the screen output: "Compton and pair events in histogram" are your events in your ARM and energy window.
- Do the math!

12.3 Generate a response for Bayesian event reconstruction

The Bayesian Compton and Electron Tracking (BCT & BET) algorithms work reliably and significantly out-perform all previous types of Compton reconstruction algorithms: More photons are reconstructed into the photo peak and background can be better suppressed.
However there is a price to pay - a significant increase in complexity: Firstly one has to generate a large data-space, based on on hundreds of millions of simulated events and secondly the event reconstruction takes almost 10 times longer, due to the large amount of absorption probabilities, which have to be calculated.

12.3.1 Generation of the data space

First, one has to generate the data spaces, based on which a quality factor for each possible event combination is calculated via Bayesian statistics. This quality factor corresponds to the probability that the events is good, based on the given measurement parameters.
The data-space has 8 dimensions, which is split into 2 six-dimensional and 1 five dimensional space, to take into account independence of some variables and to simplify its generation process.
This process splits into three parts:

1. Simulation of a significant amount of photons
2. Filling the data spaces
3. Potentially merging data spaces of different runs

12.3.1.1 Simulation The simulation should resemble the real measurements as close as possible. However, the response generator can currently only handle photons reliably. For a low-earth orbit satellites, a simulation where 10% of the photons come from the upper hemisphere and 90% from the lower hemisphere should do.
Please consider to make large enough simulations to fill the data-spaces with decent statistics. When the *.compton.good.rsp response file (BCT) or the *.central.good.rsp (BET) has an average statistics of 100 counts per bin (details in 12.3.1.3) than you have reached the minimum requirements. Depending on you geometry this might mean to simulate 500 million events and more. Also take into account that simulating the data is the fastest part, filling the data-space and therefore calculating all absorption probabilities takes 20+ times longer!
Remarks for MGGPOD users - for Cosima (Geant4) everything works per default:

- It is necessary to make the simulation with the INIT2 and ACT2 keyword
- Since you are probably running dozens of simulations in parallel, make sure to always use a different set of random numbers!
- Set the electron cut off in the input.dat file to 10^{-7}.
- You have to convert the fits file into the sim file with the special option "-s 21" of ConvertMGGPOD. This retrieves significantly more information from the fits file, so that the sim-file will definitely get *larger* than the fits-file!

12.3.1.2 Filling the data space The filling of the data-space is done with the program ResponseGenerator. Its mandatory options are:
-g: geometry file name
-f: sim file name
-r: output response file name
-m: modes: t (= track) or c (= Compton)
-c: revan configuration file

It is very important to specify the options "-m c" (or "-m t" for a tracking response) and the revan configuration file. The revan configuration file contains all options, with which the events are reconstructed. All relevant options can be set via the revan GUI. Simple press the menu option "Save configuration" after you have configured revan
The output of ResponseGenerator are several *.rsp files with represent several aspects of the response.

12.3.1.3 Merging response files You probably have done dozens of simulations in parallel, generated hundreds of GB of simulation data and now want to merge all sets of *.rsp files into one set. This is done via the program ResponseManipulator:
user@earth:~/MEGAsoft/MEGAlib> `ResponseManipulator -f New.blabla.rsp -a A.rsp -a B.rsp`
This has to be done for all different response files.
If the call "ResponseManipulator -s BaseName.compton.good.rsp" tells you that you have a statistics of at least 100 cts/bin, then you are done!

12.3.2 Event reconstruction

Now load the revan configuration file *Menu: File → Load configuration* , switch to the reconstruction *Menu: Reconstruction → Selection of algorithm* and load the *.rsp files *Menu: Reconstruction → Sequencing options* . Here, you only need to give the *.goodbad.rsp file. All other files are expected to have the same base name and to reside in the same directory as the main file.
During the event reconstruction the most likely event sequence is determined as well as an overall quality factor of the event. The quality factor represents the probability that the event is good (correctly reconstructed and completely absorbed) based on the given measurements.
This quality factor (CQF) is the main event selection criteria for mimrec. A CQF of 0.1 means that the event is good based on the measurements with a probability of 90%.

12.3.3 Additional warnings

- Whenever you change your detector, you need to redo the whole analysis!
- Always do a test with a small file first, before you start to generate 100+ GB of data!

12.4 Generate a response for the earth horizon cut

Use the program ResponseGenerator on a sufficiently large source data set. The simulation has to be done with full simulation information (ACT2 and INIT2 in case you used mgeant/mggpod).
The call is:
user@earth:~/MEGAsoft/MEGAlib> `ResponseGenerator -g Bla.geo.setup -f Bla.sim -r Bla -m e -c Revan.cfg -b Mimrec.cfg -s 10000`
The mimrec configuration file should be reasonably similar to what is used for the sensitivity calculations - of course with exception of the earth horizon cut.
When a call to ResponseManipulator, tells you that the average bin content is well above 100, then you have enough statistics:
user@earth:~/MEGAsoft/MEGAlib> `ResponseManipulator -s Bla.compton.ehc.rsp`
You can view the content of the Response Matrix via:
user@earth:~/MEGAsoft/MEGAlib> `ResponseManipulator -v Bla.compton.ehc.rsp x y 0 0 0 0 0`