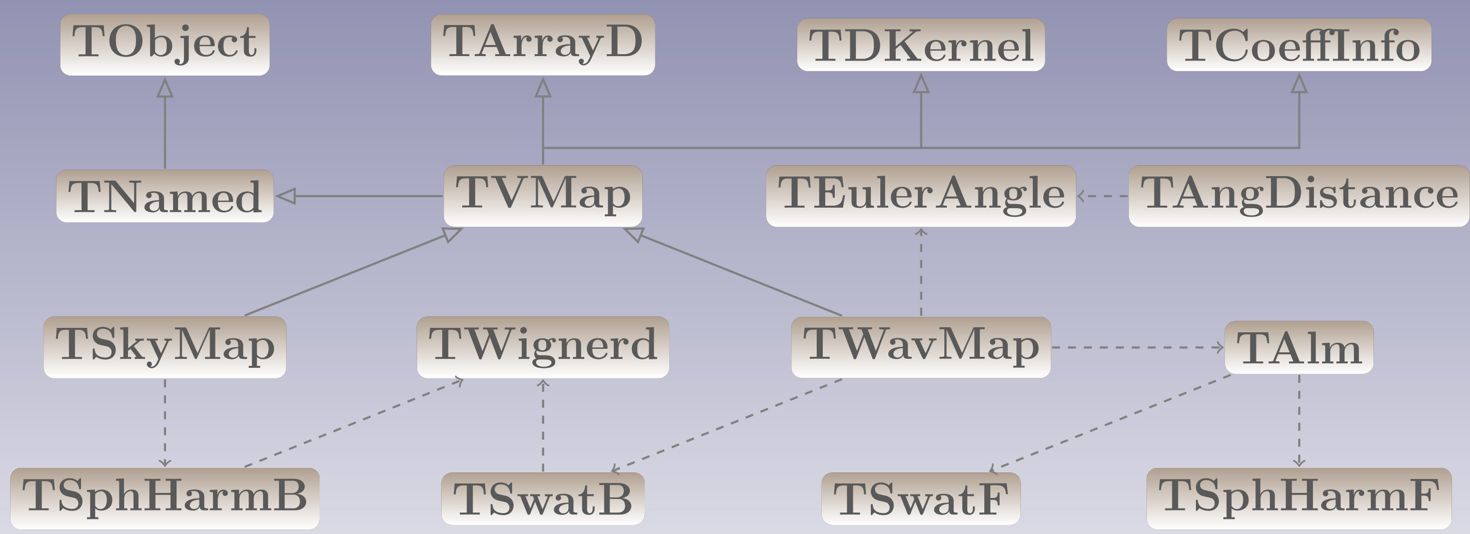
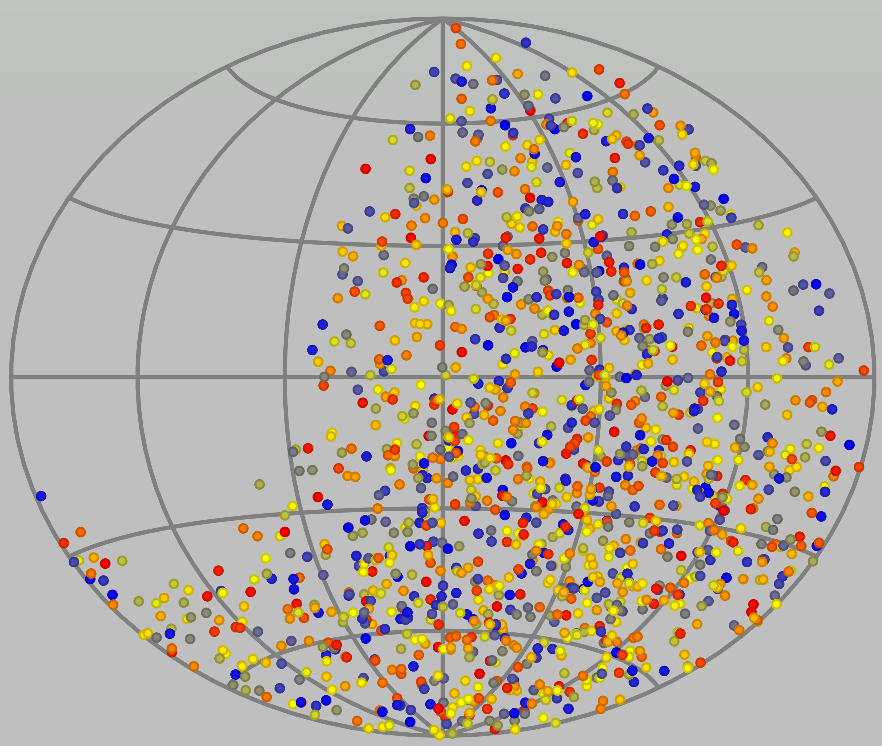


SWAT

The Spherical Wavelet Analysis Tool



Manual for version 1.53 - Marcelo Zimbres

Contents

1 Introduction

SWAT is a package for analysis of functions that are defined on the sphere. It has been used extensively to analyze data collected by the Pierre Auger experiment, CRPropa simulations and other simulation data. The program has been written from scratch as part of my my Ph.D, where it evolved from a simple ROOT macro and got bigger and bigger, since the code may be useful to other people I decided to organize, document it and make it public. The project main feature is its C++ implementation of the spherical wavelet transform as presented in [2]. Some attractive features of SWAT are:

- Harmonic and Wavelet transform on the sphere.
- Interface to Healpix code, which is included in the build system (the user does not have to install it).
- Easy selection of events hitting sky window.
- Calculation of deflection vs. $1/E$ graphs.

- ROOT and FFTW are the only prerequisites.

The package includes some parts of the Healpix code. There are two reasons why I decided to include it here, instead of just link against Healpix libraries.

1. I do not need all Healpix routines and support to the fits format.
2. I usually need shared libraries to call Healpix code in a ROOT session, which are not built by Healpix build system since most of its code are C++ templates.
3. Healpix installation used to be messy.

Most of the theoretical details of analysis of functions defined on S^2 and $SO(3)$ were taken from [2] and references therein.
The code has been tested in LINUX and MACBOOK, but the code is portable enough to be built on other platforms. In the following we describe how to use SWAT. Questions concerning the software can be sent to Marcelo Zimbres mzimbres@gmail.com

Acknowledgements

Most of SWAT were developed with financial support from CAPES. Additionaly I would like to acknowledge the Institute of Physics at UNICAMP and the Bergische Universität Wuppertal for finacial support, Brunel University for granting me a scholarship to participate in the Cern School of Computing, the Ettore Majorana Foundation and Center for scientific Culture, the brasilian group of the Pierre Auger experiment, with special thanks to my supervisor Ernesto Kemp, for beliving in the project and Rafael Alves Batista for testing the code. The Pierre Auger Group at university of Wuppertal, with special thanks to Nils Niertenhöfer for helping me with CRPropa.

1.1 Installation

As a prerequisite to install SWAT you need ROOT and FFTW installed, the configure script will fail if they are not installed. SWAT uses autotools to generate the configure script and the Makefile, so you can expect all the standard configure options and makefile targets. The lines bellow will install the libraries on "/usr/local"

```
$ tar -xvzf swat-1.53.tar.gz
$ cd swat-1.53
$ ./configure CXXFLAGS="-O3 -ffast-math"
$ make
$ sudo make install
```

The flags passed to the configure script are optional, but greatly improve performance. To use some of the features of the package, I am assuming you are able to build shared libraries on your platform. To easy the task of loading swat libraries the macro load.C, will be installed on "/usr/local/share/swat". To load the libraries in ROOT's C++ interpreter you have to execute it in your ROOT session, or add it to your code .rootlogon.C macro.

```
$ cp /usr/local/share/swat/load.C ~/.rootlogon.C
$ root # The .so's are automatically loaded here.
```

All SWAT classes are now available in the ROOT session with syntax highlighting (currently I do not use swat from inside a root session, but it may be usefull for others). You should also be able to generate documentation in HTML format with the macro "prefix/share/swat/makehtml.C". You have to run this macro from the directory where you built SWAT. The documentation will be built in the directory htmldoc, this is a nice way to get aquainted with the code (this is meant for those wanting to develop).

1.2 Getting ready

To use Herald file (a file distributed by the Pierre Auger collaboration containing collected data), you will have to convert the ascii file to a **TTree** and save it in a .root file, for that you should use the macro prefix/share/swat/convert_herald.C where prefix is usually /usr/local. Copy this macro to the same directory where you have the herald data file. The macro will read a file with name "herald.dat", so you may have to rename your file or edit the macro. For CRPropa simulations you can configure CRPropa to output a .root file instead of a text file. The code uses the title of the **TTree** to differentiate between a Herald and CRPropa file. The title of the TTree for CRPropa must be "CRPropa 3D events", as far as I know, this is the default.

1.3 High quality graphs with pgfplots

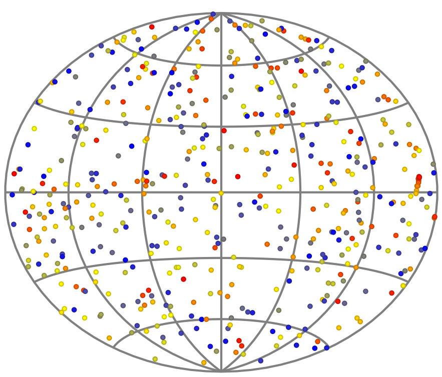
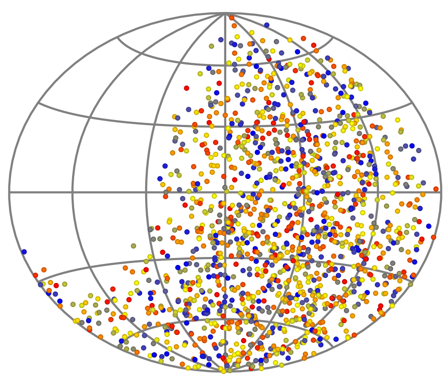
If you have pgfplots installed on your machine, you can use the LaTeX files which are available in the directory pgfplots in swat root dyrectory, to generate high quality graphs using some output of swat analysis. You may have noted that after running swat_find you get many text files in your working directory. They are input files for the LaTeX files. In the following subsections we show how to use the material.

1.3.1 Skymaps

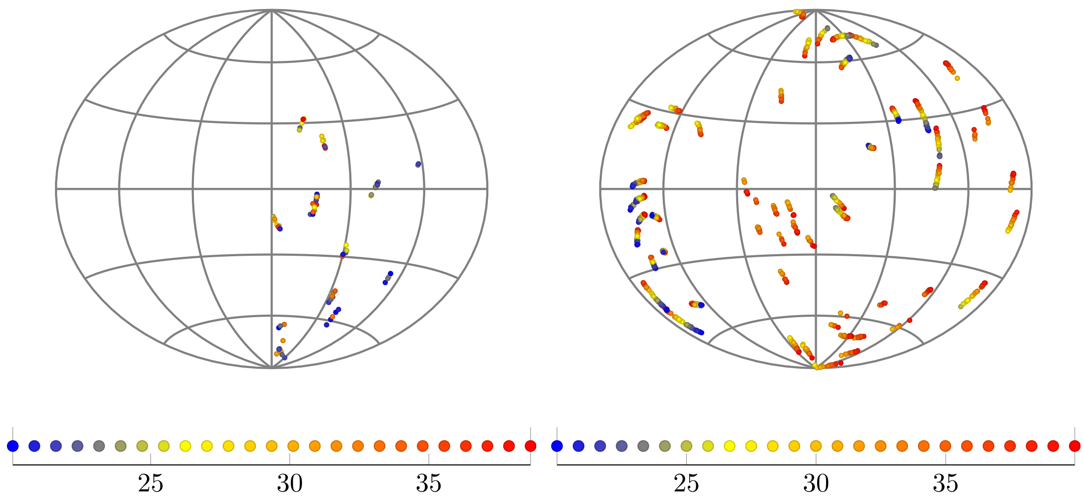
When you run the program swat_find or swat_sim, they produce a file named *skymap.dat*. This file contains the coordinates and energy of the events that passed the cut. To generate a graph for an isotropic sky for example, one can use:

```
$ tar -xzf skymap.tar.gz
$ cd skymap
$ swatsim -n 1000 -s 1
$ pdflatex --jobname=skymap-f1 skymap.tex
```

Example skies can be seem here:

The program will also generate the file mult_cand.tex, this file contains only the events which hit the tangent plane in the position found by the wavelet analysis. An example can be seem below on the left side. On the right side we see a skymap simulated with CRPropa.



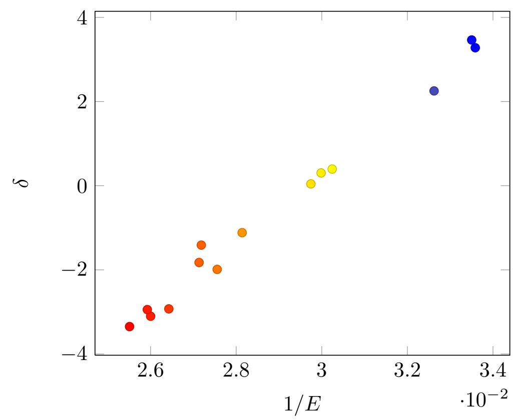
Usually the energy of events is represented in skymaps as circles of varying size, where events with higher energy have larger circles. I do not like such representations since they can give the false impression that the angular resolution gets worse as the energy increases, which is not true. Additionally, these circles pollute the figure. In the representations I am using, colors are used to differentiate energies which in my opinion is a much clear way of representing the sky.

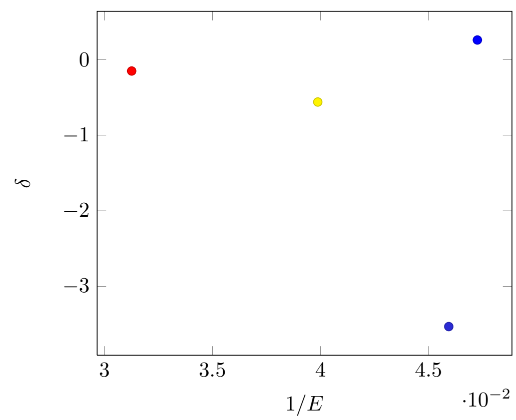
1.3.2 Energy deflection graphs

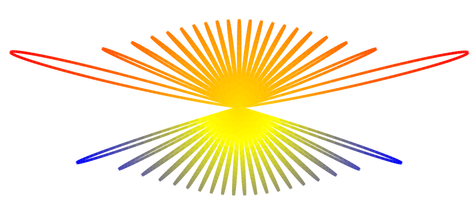
In addition to skymap.dat, the files corr_graphN.dat will be also generated, with N varying from 0 to 15 (this number can be passed in the command line). These files contains the energy-deflection graphs, of the events the hit the tangent plane in the position found by the wavelet analysis. On figures 1.3.2, we see two examples. The graph on the left originates from a CRPropa simulation.

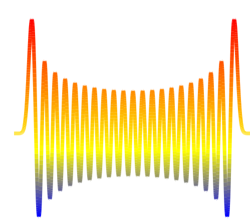
1.3.3 Wigner-d functions

Swat implements the calculation of the wigner d^l_{mn} functions via FFT. You can use the file pgfplots/wignerd.tar.gz to generate some nice graphs.


4
2
0
−2
−4
δ
2.6
2.8
3
3.2
3.4
1/E
·10−2


0
−1
−2
−3
δ
3
3.5
4
4.5
1/E
·10−2





2 Transforms on the Sphere

Most SWAT functionality is based on Fourier and wavelet transforms on the sphere, therefore we give in this section a brief overview of analysis of functions on the sphere. Before that however, it is important to mention the problem of sampling functions on the sphere, which is refered to as its pixelization. Due to the fact that most of the time we work with function defined on the line (temporal series, for example) or on the plane (images in general) we do not have to care much about how to sample. We simply divide it in squares of same area and everything works just fine. However, when we try the same thing on the sphere we imediately see that it is impossible to achieve same area pixelization on the sphere due its curvature.
There are two main features we would like to have when dividing a sphere in pixels. First we would like to have pixels of same area, this is achieved by the Healpix pixelization 2. Second, we would like to to use FFT to calculate functions on the sphere instead of slow recursion and that is achieved by ECP Pixalization, which stands for "Equidistant cylindrical projection"2. Unfortunately theses two features can not be achieved at the same time. In this software we decided to use ECP due to its simplicity. In ECP, the three Euler angles, which we are using to parametrize SO(3), are sampled as follows α, β and γ

$$\alpha_{j_1} = \gamma_{j_1} = \frac{2\pi j}{2B}, \quad \beta_{j_1} = \frac{\pi(2k+1)}{4B} \tag{1}$$

2.1 Fourier Transform on SO(3)

A function $f \in L^2(SO(3))$ can be decomposed as follows

$$f(\alpha,\beta,\gamma) = \sum_{l=0} \sum_{m=-l}^{l} \sum_{n=-l}^{l} f^l_{mn} D^l_{mn}(\alpha,\beta,\gamma). \tag{2}$$

where

$$D^l_{mn}(\alpha,\beta,\gamma) = e^{-im\alpha} d^l_{mn}(\beta) e^{-in\gamma}. \tag{3}$$

is called "Wigner-D functions" and form an irreducible representation on SO(3). The functions d^l_{mn} are called "small wigner-d functions".

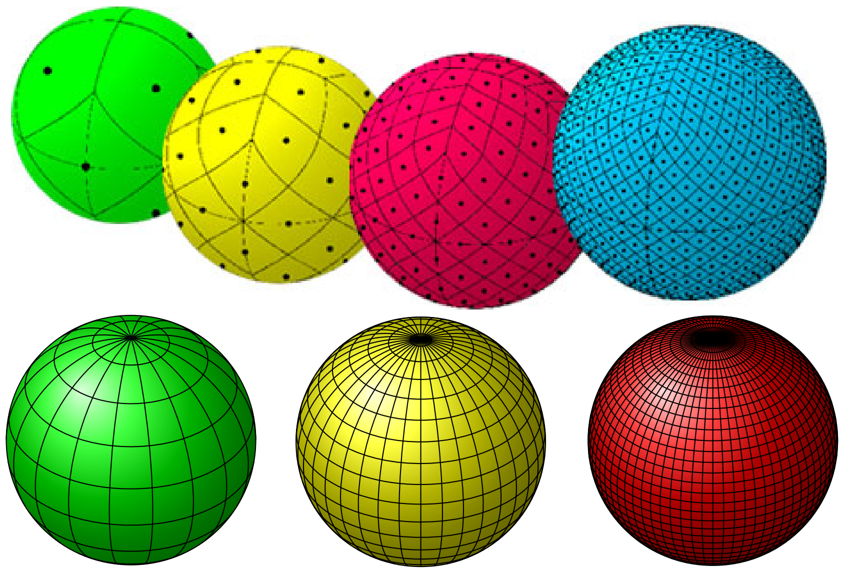


Figure 1: Above:Healpix pixelization of the sphere. Below: The ECP pixelization.

The inverse of 2 on ECP pixelization is given by

$$f^l_{mn} = \frac{1}{(2B)^2} \sum_{j_1=0}^{2B-1} \sum_{j_2=0}^{2B-1} \sum_{k=0}^{2B-1} w_B(k) f(\alpha_{j_1}, \beta_k, \gamma_{j_1}) D^{l*}_{mn}(\alpha_{j_1}, \beta_k, \gamma_{j_1}) \tag{4}$$

where B is the band limit and the quadrature weights are given by

$$w_B(k) = \frac{2}{B} \sin\left(\frac{\pi(2k+1)}{4B}\right) \sum_{j=0}^{B-1} \frac{1}{2k+1} \sin\left((2j+1)(2k+1)\frac{\pi}{4B}\right) \tag{5}$$

where $0 \leq k < 2B$.

3 Installed programms

In the following sub-sections we will describe the main programs which SWAT installs and which provide an easy interface to the wavelet analysis.

3.1 swat_find

swat_find is the main program. It can be used to find sources in Pierre Auger data and CRPropa simulations. Beyond that, it can be also used to

test whether the events belong to a multiplet, calculating energy-deflections graphs for the sources found. All information is saved in .root format. A source here is a term used to mean something with a position on the sky, usually described by (θ, ϕ) and an orientation. We will use the three Euler angle (α, β, γ) to describe the source.
It uses the following algorithm:

1. The TTree containing Pierre Auger data or CRPropa simulations is read from the file passed in the command line with option -f.

2. Events with energy in the range $emin < E < emax$ are selected and used to fill a Healpix map. The options -i and -e are used to pass the energy range.

3. The Healpix map generated is transformed to wavelet space. The parameter N, passed by the option -N gives the precision on the ability of the wavelet to find the angle γ, which can range from 1 - 128 in our implementation. The precision in degrees are calculated with the formula $180/N$ The scale on which the analysis is performed is passed with option -j. It ranges from 0 - 8. We found out that the best results are achieved with j = 1.

4. A partial sort is used to find the 15 largest wavelet coefficients.

5. The euler angles found will be used to calculated the tangent plane equation for each source. The width and length of the plane are passed with options -w and -l respectively.

6. A loop on the data is made to select all events hitting the tangent plane. The energy cut is used again.

7. The energy-deflection graphs are calculated and the correlations found are printed on the screen.

All the information is saved in the file "sources.root". Lets see an example:

```
$ swatfind -j 1 -N 127 -i 20 -e 40 -w 2 -l 10 -f chain.root
TFile**          chain.root       CRPropa output data file
 TFile*          chain.root       CRPropa output data file
  OBJ: TNtuple   events   CRPropa 3D events
```

```
OBJ: TEventList        list     20 < emin && emax < 40
OBJ: THealpixMap       hmap     Healpix sky map
OBJ: TGraph    g0      C = -0.993, N = 4
OBJ: TGraph    g1      C = -0.986, N = 5
OBJ: TGraph    g2      C = 0.000, N = 0
OBJ: TGraph    g3      C = -1.000, N = 2
 ...
```

The N in the TGraph title is the number of events in the graph (or the number of events which hit the tangent plane) and is not to be confused with the N passed by command line option -N. C is the correlation of the energy-deflection graphs.
Now, if you want to see the exact location of the source:

```
$ root sources.root
root[1] .ls
TFile**          sources.root
 TFile*          sources.root
  KEY: THealpixMap       hmap;1   Healpix sky map
  KEY: TEulerAngle       source0;1
  KEY: TEulerAngle       source1;1
  KEY: TEulerAngle       source2;1
root [2] source0->Show(1)
wav(137.812,-27.4219,159.638) = -0.035312
root [3]
```

Help description

```
Searches for multiplets in herald data or CRPropa
simulations. Calculates correlation of events hitting
stripe on the tangent plane and dispalys on the screen.
The results are saved to sources.root file. Two input
file formats are supported, both are TTrees saved in a
.root file. The TTrees can be either the output of
CRPropa or a Herald file converted to TTree (see macro
macros/convert_herald.C in swat source tree).

Usage: swat_find [ -j scale] [-N number] [-i emin]
       [-e emax] [-w width] [-l length] [-f file.root]
       [-n nsources] [-t wav_threshold]

Options:
```

```
-h:      This menu.
-j:      Wavelet scale. It is a number in the range
         0 <= j <= 8, defaults to 1.
-N:      Band limit of wavelet, in the range 0 < N <= 128,
         defaults to 1.
-i:      Minimum energy of events, defaults to 20 EeV.
-e:      Maximum energy of events, defaults to 40 EeV.
-w:      Width of tangent plane, defaults to 2 degrees.
-l:      Length of tangent plane, defaults to 10 degrees.
-f:      Root file containing Tree with data, defaults to
         chain.root.
-n:      Number of sources to look for. Default to 15
-t:      Wavelet threshold value.
```

3.2 swat_sim

swat_sim uses Monte Carlo simulations to calculate the probability of a multiplet happen by chance on a isotropic sky. The distribution of E, θ and ϕ must be passed in the command line, it can be generated by swat_gen The probability is calculated as follows:

- The program simulates n isotropic skies. If you want to add your simulated events to each sky you can use the option -f and a TTree in CRPropa format will be read and added in each sky.
- For each sky simulated the analysis made by swat_find is used to calculate the correlation coefficient.
- If the correlation is larger than C , passed with option -c and has more than n events, passed with option -m, then the multiplet has passed the criteria.
- The probability will be total number of multiplets passing the criteria divided by number of skies simulated.

Beyond the probability, the program outputs two additional histograms:

1. A histogram of the correlation coefficients found, for which the total number of events is larger than the value passed with -m.
2. A histogram of the number of events in the tangent plane.

3. A histogram of the largest wavelet coefficients found.

Help message

```
Calculates the probability of a multiplet with minimum
correlation c > c_0 (see -c option), minimum number of
events m > m_0 (see -m option) and where the magnitude of
the wavelet coefficint e C > C_0 (see -C option), happen
by chance using wavelet analysis.  First an isotropic sky
is simulated (the coverage and energy distribution must
be provided) and the wavelet representation of the sky is
calculated, the euler angles of the largest coefficient
is used to calculate the equations of the tangent plane
at the position found (the euler angles). The correlation
c is calculated including all events that hit the tangent
plane, whose size is specified with the options -l and
-w. The probablility will be the number of multiplets
with c > c _0, C > C_0 and m > m_0, divided by the number
of skies simulated. Additionaly, seven other quantities
are calculated:

   1 - The histogram of the number of events that hit the
       tangent plane.
   2 - The histogram of the c's found for which the
       number of events is greater than m_0 and C > C_0
       (passed in the command line).
   3 - The histogram of the magnitude of wavelet
       coefficients C_0.
   4 - The histogram of the mean of wavelet coefficients.
   5 - The histogram of the variance of wavelet
       coefficients.
   6 - The histogram of the skewness of wavelet
       coefficients.
   7 - The histogram of the kurtosis of wavelet
       coefficients.

If -f option is used, a TTree in the file will be read
and events will be added to the analysis, this is useful
to include a simulated multiplet on the analysis, hiding
it in the isotropic backgroung the test the algorithm.

It is also mandatory to specify:
```

```
  − Energy distribution.
  − The theta distribution.
  − The phi distribution.

This is the distribution the background events have to
follow. These distributions are read from a root file.
Use swat_gen and swat_coverage to generate them.

Usage: swat_sim [ −j scale] [−N number] [−n nevents]
       [−s skies] [−i emin] [−e emax] [−c corr] [−m mevents]
       [−w width] [−l length] [−f file.root] [−C min_wav]
       [−d energy] [−a coverage]

Options:

−h:     This menu.
−j:     Wavelet scale, a number in the range 0 <= j <= 8,
        defaults to 1.
−N:     Band limit of wavelet, in the range 0 < N <= 128,
        defaults to 1.
−n:     Number of events in the simulated sky, defaults
        to n = 1000
−s:     Number of skies to simulate, defaults to 100.
−i:     Minimum energy of events, defaults to 20 EeV.
−e:     Maximum energy of events, defaults to 40 EeV.
−c:     Minimum correlation, defaults to 0.2.
−C:     Minimum Magniftude of wavelet coefficient,
        defaults to 0.0.
−m:     Minimum number of events hitting tangent plane.
−w:     Width of tangent plane, defaults to 2 degrees.
−l:     Length of tangent plane, defaults to 10 degrees.
−f:     Add events in TTree stored in file to the
        simulated sky.
−d:     Histogram with energy distributions.
−a:     Histogram with theta and phi distributions.
```

3.3 swat_gen

Generates distribution of E θ and ϕ coordinates that will be used by swat_sim program. Two kinds of distribution are supported, isotropic or Auger distribution if herald data is provided.

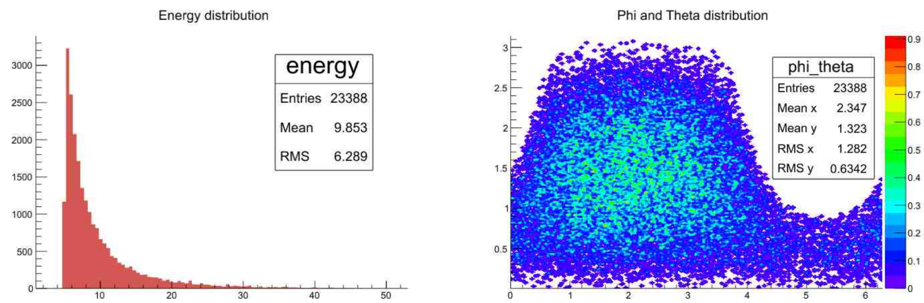


Figure 2: Energy distribution on the left. On the right angular distribution. Both for Auger experiment.

3.4 swat_prob

```
Reads a file output by swat_sim and calculates
probabilities.  Use the macros cprob.tex and
wprob.tex in pgfplots directory to produce
probability graphs.

Usage: swat_prob [-n n_steps] [-o outpu-file] [-t]
                 [-k kind]

Options:

  -h:      This menu.
  -o:      Output file, where probabilities
           will be recorded.
  -t:      Calculates P(q >= q_0) if provided,
          P(q < q_0) otherwise.  q will be the
           correlation or wavelet coefficient,
           depending on the value passed in option -k
  -k:      Either 1 for correlation or two for
           wavelet coefficient.
  -n:      Total number of steps (points on
           probability graph). Defaults to 10.
```

3.5 swat

This program is only used to benchmark and test the algorithm. It can test both the spherical harmonic transform and the spherical wavelet transform.

```
$ time swat -J 8 # Will test spherical harmonic transform.
$ time swat -J 8 -N 3 # Will test wavelet transform.
```

Help message

```
Tests the algorithm performing forward and backward
transform.  Both spherical harmonic transform (if option
-N is not provided) or spherical wavelet transforms can
be performed. I use this program to benchmark my code
using the time command:

$ time swat -J 8 -N 127

for example. If forward folloed by backward transform do
not result in the data, with precision 1e-10, program
exits with EXIT_FAILURE status. For example

$ swat J8 -N 127
$ echo $?
0
$

Usage: swat [-J j] [-N n]

Options:
-h:     This menu.
-J:     Sets band limit of the signal to 2^J, defaults to
        J = 7.
-N:     Band limit of wavelet to be used.
```

References

[1] J Fourier Anal Appl (2008) 14: 145179.

[2] Mon. Not. R. Astron. Soc. 000, 122 (2007).

[3] http://www.ifi.unicamp.br/~mzimbres/