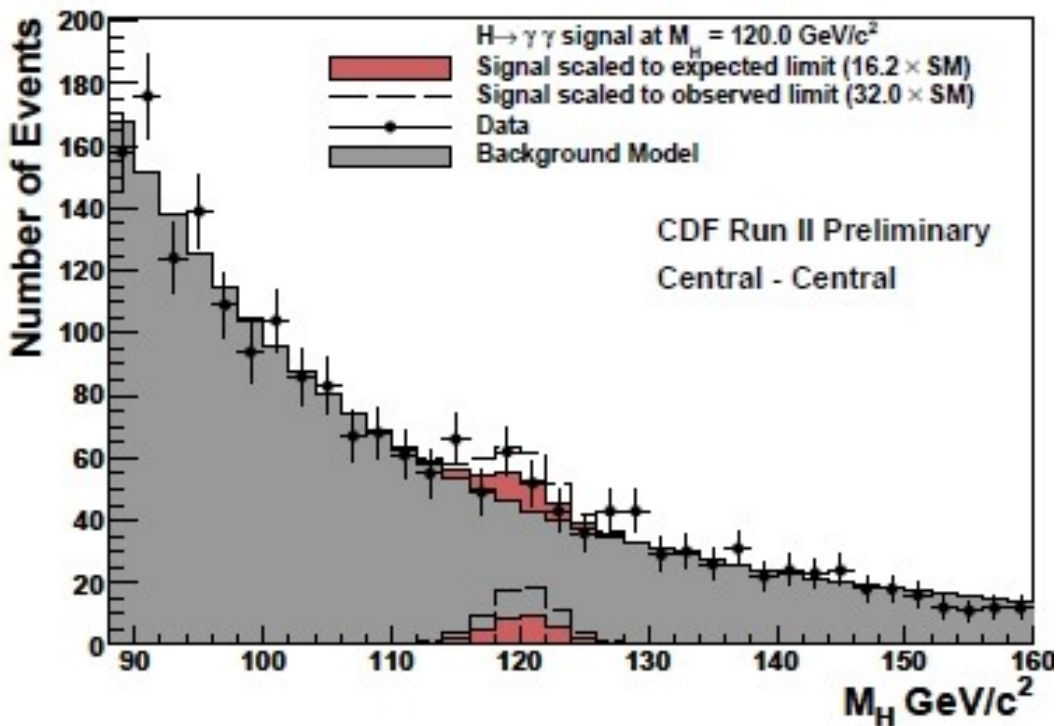


## Spectral analysis

In experimental particle physics it is common the need of distinguishing a signal from the background. A “standard cut analysis” is used whenever this need is fulfilled by the mean of selections on a set of observables until the background contamination in the selected sample becomes negligible. A “spectral analysis” can be used when the background cannot be reduced or when a high selection efficiency is needed.

In a spectral analysis signal and background from experimental data are estimated together by studying the distribution of the observable for which the best separation between the two components can be reached. A fit of this convoluted distribution is performed once the shape of the two independent distributions of signal and background is known (usually by the mean of simulations).



Example of spectral analysis

## Goal of these exercises

**The aim of these exercises is to find out which is the number of signal events in a given data sample by fitting the event distribution of an observable with a function defined as the sum of the signal and background distribution functions.**

Basic informations: you will work on a set of events coming out from a PAMELA experiment simulation. The given file

```
/home/mocchiut/pamela/data/pamelasimu.root
```

contains the TTree `pamcalotree`, storing data with the PamCalo class, header file:

```
/home/mocchiut/pamela/PamCalo/inc/PamCalo.h
```

so library:

```
/home/mocchiut/pamela/PamCalo/lib/Linux/libPamCalo.so .
```

The ROOT file contains about 5.700.000 events: protons, electrons and positrons mixed together in an energy range from 10 to 300 GeV.

Positrons to electrons ratio in this file is about 0.1 . Electrons to protons ratio in this file is about 0.007 .

**In these exercise, consider only the energy range [25-30] GeV and the distribution of observable “ $q_{tt} \equiv q_{tr}/q_{tot}$ ”. Events must be “pre-selected” using selection cuts on the “ $q_{tot}/energy$ ” distribution (energy-momentum match). Signal-like events are represented by electrons (`pID==1`), background-like events are represented by protons (`pID==0`), experimental data events are represented by protons and positrons in the file (`pID==0 || pID==2`).**

# Exercise 1

Determine the pre-selection cuts:

using CINT/ROOT, open the data file

```
/home/mocchiut/pamela/data/pamelasimu.root
```

and plot the `abs(qtot/energy)` distribution for signal like events (`pID==2`) in the energy range between 25 and 30 GeV. Take note of the “Mean” and “RMS” of the distribution; events will be selected (look at exercise 2) if `qtot/energy` is greater than “Mean-1.2\*RMS” and lower than “Mean+1.2\*RMS” (hint: you can approximate the values to the closest integer). Save the histogram as a jpeg file and write in the README file the Mean, the RMS and the commands used to obtain the plot.

## Exercise 2

Write an executable compiled program which reads the input file

```
/home/mocchiut/pamela/data/pamelasimu.root
```

and gives as output a new ROOT file containing a TTree with two variables (a TBranch for each one):

- pID
- qtt

where qtt is defined as qtr/qtot.

Save into the new file events which satisfy both the following two conditions:

1. the event lays in the energy range 25 – 30 GeV (hint: pay attention to the sign of “energy”! use “fabs” function);
2. the so-called energy-momentum-match is between “Mean-1.2\*RMS” and “Mean+1.2\*RMS” MIP/GeV, where Mean and RMS are the results of exercise 1 (hint: `fabs(pc->qtot/pc->energy) > ... && fabs(pc->qtot/pc->energy) < ...`, where `pc` is an object of the class PamCalo; if unable to complete exercise 1 use the range [180.-310.]).

Hints:

- to compile, remember to add also the compilation flags:

```
-I/home/mocchiut/pamela/PamCalo/inc
```

```
-L/home/mocchiut/pamela/PamCalo/lib/Linux/
```

```
-lPamCalo
```

- to run, remember to export LD\_LIBRARY\_PATH:

```
export LD_LIBRARY_PATH=/home/mocchiut/pamela/PamCalo/lib/Linux/:$LD_LIBRARY_PATH
```

- the output file should have a size of about 12K, if you have quota problem you can write the output on the linux temporary directory “/tmp”.

## Exercise 3

Write a ROOT-CINT script which reads the output file of exercise 2 (should be similar to this one: `/home/mocchiuti/scripts/EM_output_220113.root` use this file if you are not able to complete or run exercise 2) and gives as output on the screen and on the disk (pdf format) a TCanvas divided into three pads (one column, three rows – hint: `TCanvas::Divide`) which contains from top to bottom (use TH1D with the same binning):

- the signal-like events distribution ( $p_{ID}=1$ ) histogram fitted by a Gaussian function (3 free parameters,  $f1=G(N1,\mu1,\sigma1)$  ), draw the fit line as well;
- the background-like event distribution ( $p_{ID}=0$ ) histogram fitted by another Gaussian function (3 free parameters,  $f2=G(N2,\mu2,\sigma2)$ ), draw the fit line as well;
- the experimental data event distribution ( $p_{ID}=0 || p_{ID}=2$ ) histogram fitted by a function which is the sum of two Gaussian functions that have the mean value fixed to the mean values and sigmas of the previous two fit (that is: 2 free parameters:  $f3=G(N3, \mu = \mu1, \sigma = \sigma1) + G(N4, \mu = \mu2, \sigma = \sigma2)$  ) (hints: look at examples in the TF1 reference manual, look at method `TF1::FixParameter`). Draw the fit line and (with a different line style) its two Gaussian components separately.

Answer the following questions:

1. How much is the integral (S) of the signal-like fitted function (hint: `TF1::Integral`)? This number (divide by the bin width) is a smoothed number of signal like events.
2. How much is the integral (D) of the signal experimental data distribution fit, that is integral of Gaussian  $G(N3,\mu1,\sigma1)$ ? This number (divided by the bin width) is a smoothed number of signal events.
3. What is the result of  $D/(D+S)$  ? is it an expected result?

## Preparing the output

- create a directory named with the following format:  
YourInitials\_C++2012  
(for example in my case it would be: EM\_C++2012)  
put inside this directory ALL the files you want me to correct and look at.
- ALL files names format (but Makefile, if any) must be like:  
YourInitials\_something.extension  
(for example in my case I would create files: EM\_main.cpp,  
EM\_myscript.C, EM\_OutputHistogram1.pdf, etc. etc. )
- create a README text file (named like EM\_README.txt), inside the file write:
  - **your name and surname**
  - a list of the files you are submitting
  - **in details** how to compile and run the programs
  - any other comment and answer to question(s) rised in the exercise description
- create a compressed tarfile containing the directory:

```
bash> ls
EM_C++2012
bash> tar zcf EM_C++2012.tar.gz EM_C++2012/
```
- copy the tarzipped file on the USB key I will circulate

## Timing and rules

- You have four hours time to do your work.
- You can search the web, look at manuals, look at any note you wrote during the course, etc.
- We will discuss what you have written at the oral examination on 2013/01/25, until that (if needed) you can change and improve your programs. In that case prepare an electronic version we can look at during the oral examination, we will compare it to the one handed in today and we will discuss any change and/or correction.