Introduction to RooFit

Gagan Mohanty
EHEP Group, TIFR

1 June 2011

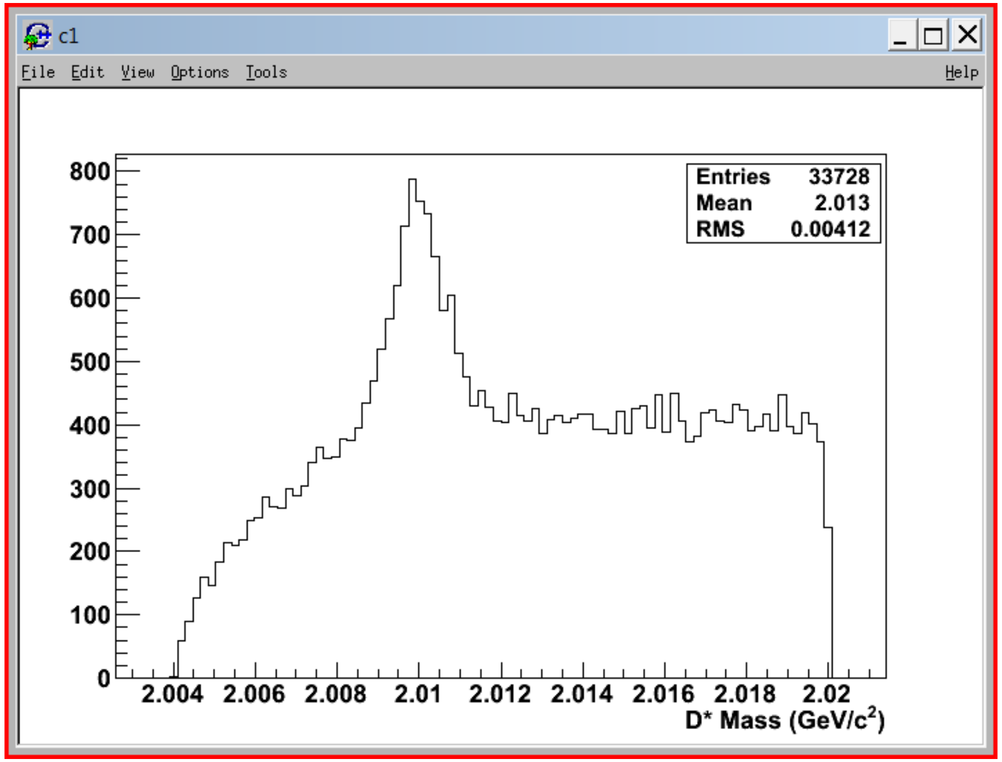
Prologue

- The story is not from an expert, rather from an experienced user
- I shall try to make this one lecture as first-hand as possible
- For somebody conversant with RooFit, the lecture may not be that useful
- In that case, I need your helps to make them useful to others
- A detailed documentation can be found at the RooFit webpage

http://roofit.sourceforge.net/

What do need the fit for?

- We will come across in life many plots or “distributions” as one given here



- We want to know
  - How many D* are there (signal)?
  - What is the width (resolution) of the mass peak?

    - We have to fit now

Fit and the jargons

- In Belle/CMS, we measure a number of "observables" x_i such as M_{bc}, ΔE, dilepton invariant mass, missing E_T and so on
- We want to determine one or more "parameters" p_i, e.g., branching fraction, mass and width of a resonance etc.
- Let's describe the distribution of our observables by a "probability distribution function" (PDF), that is a function of both the physics parameters and the observables themselves
- We choose the PDF based on the reading of the best telescope we have(!) ➡ which function will fit the distribution well?
  - The PDF is normalized over the observables: $\int dx_1 ... dx_n \; f(x_1 ... x_n; p_1 ... p_n) = 1$
  - $f(x;p)\, dx_1 ... dx_n$ is the probability that for a given event, the observables will be in the range $dx_1 ... dx_n$
  - Vary the parameters to make the PDF match with the actual distribution of the observables as well as possible, and by doing so determine the parameters

Binned *vs*. Unbinned Fits

- We all are quite familiar with a binned (least-squares) fit:
  - Bin data into a histogram and minimize $\chi^2 = \sum_i \frac{(y_i - y_{exp})^2}{\sigma_i^2}$
  - For a histogram $y_i = N_i$ and $\sigma_i = \sqrt{N_i}$ (Poisson statistics in each bin)

- In RooFit, we use a fancier (and better) technique for fitting called an "unbinned maximum likelihood fit"
  - Likelihood $\mathcal{L}(p_1, ...p_m) = \prod_i f(x_{1,i}, ...x_{n,i}; p_1, ...p_m)$
  - Vary the parameters to maximize the likelihood, or equivalently minimize

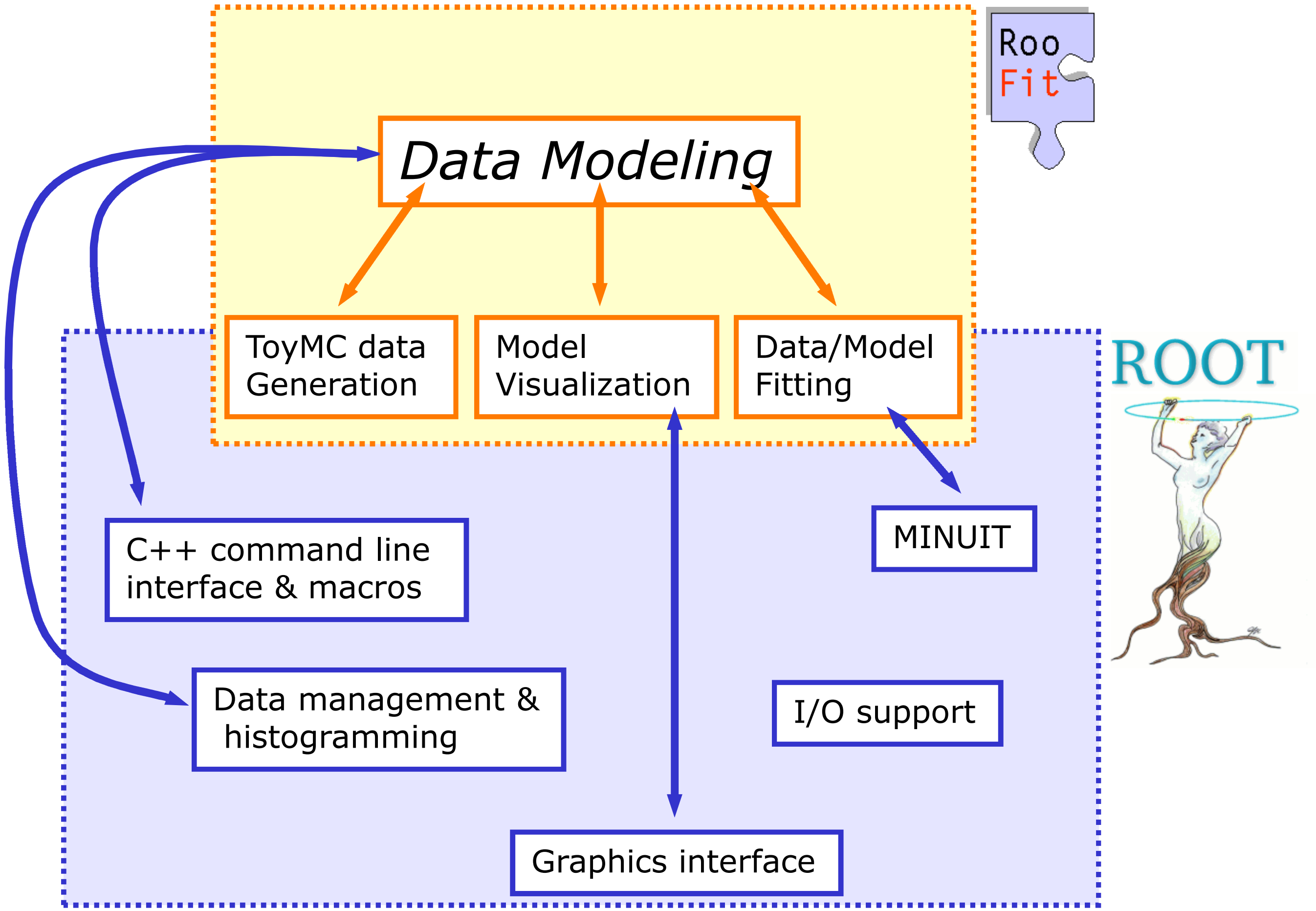
    $$-\ln \mathcal{L}(p_1, ...p_m) = -\sum_i \ln\left[f(x_{1,i}, ...x_{n,i}; p_1, ...p_m)\right]$$
  - A least-squares fit mimics a maximum likelihood fits if the bins are narrow and all bins have many events
  - As you could now see, maximum-likelihood fit is better (especially for a small dataset), but it takes longer time

What is RooFit (Fit in ROOT?)

- ROOT is a collection of C++ classes, it can do the fitting
  - Not very convenient
- RooFit, in other hand, provides more classes designed to make fitting a much smoother experience
  - One example, you need not worry about the normalization
- RooFit is not "Fit in ROOT" rather much more than that
- Above all, it is implemented as a part of ROOT (v5 or later)

Looks technical, no?

Extension to ROOT – (Almost) no overlap with existing functionality



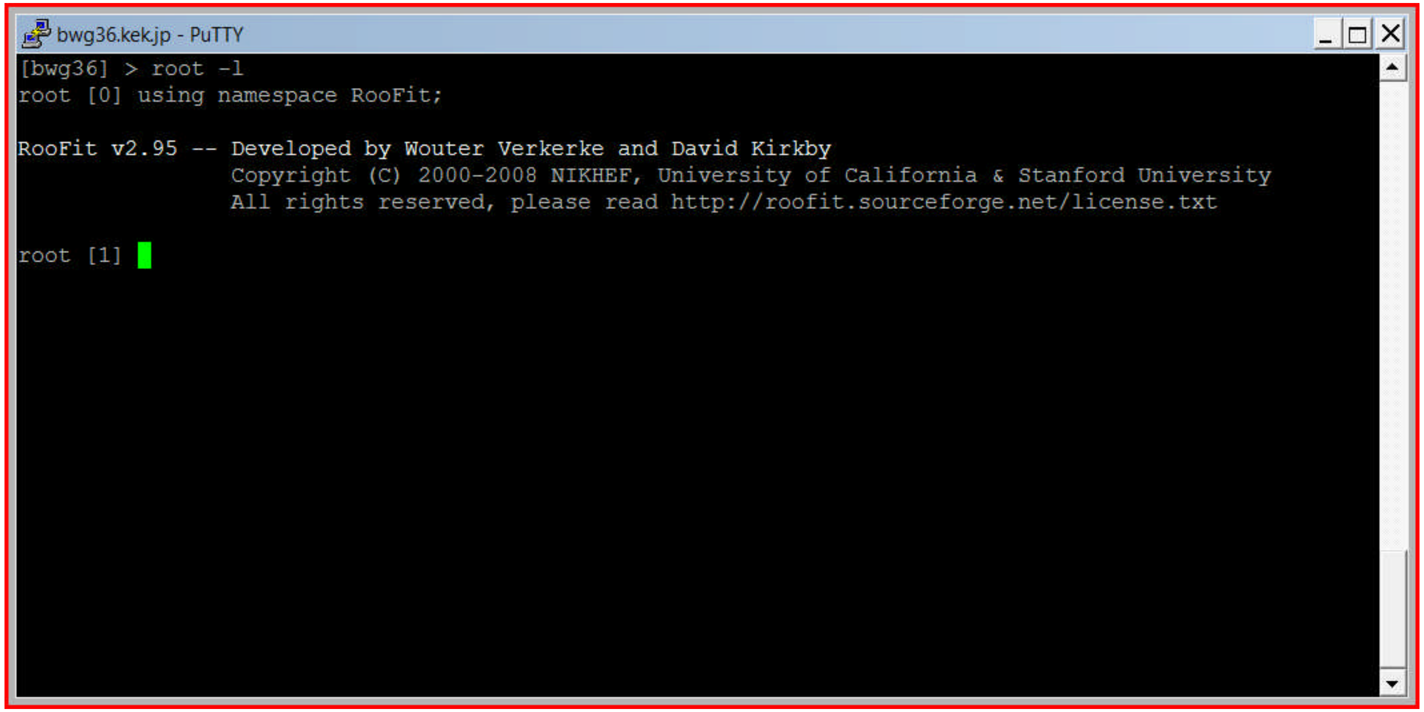
Steps to remember

- First load RooFit into ROOT
- Load data events into RooDataSet
- Apply selection requirements, or cuts (if needed)
- Define PDF(s)
- Do the fit   (you can generate toy and fit here)
- Extract out the physics parameters
- Make a nice plot

Steps for invoking RooFit

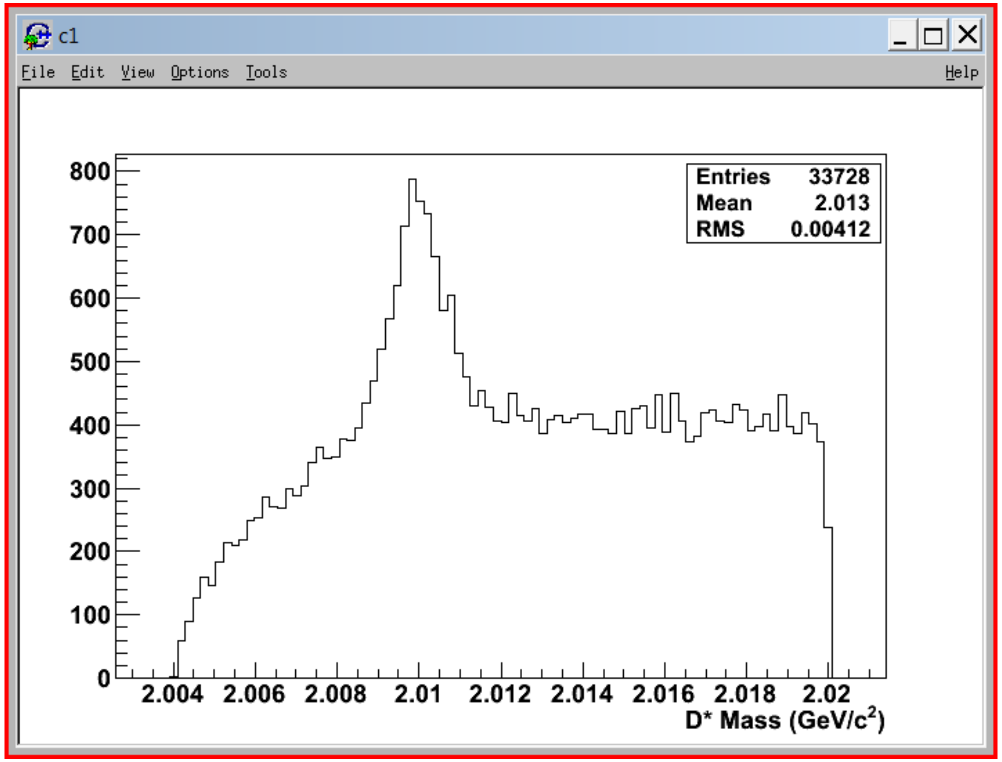
- First make sure you have correct setup for ROOT
  - See myscript at http://www.tifr.res.in/~gmohanty/RooFit

- Invoke a ROOT session ➡ root -l
- Then RooFit within ROOT ➡ using namespace RooFit;



Let's go back to slide# 2

- We want to know
  - How many D* are there (signal)?
  - What is the width (resolution) of the mass peak?



- We shall use RooFit to accomplice this task

Files used in this tutorial

- Ntuple file mydsks-on.root and macro with RooFit commands f1.cc are at http://www.tifr.res.in/~gmohanty/RooFit/

- Shall go through the macro slowly but steadily

- Any question you have stop me; no need to wait till the end

- If you are a fan of PAW and HBOOK, you should realize that RooFit works with the ROOT framework

- It's quite self-sufficient in a way and can deal with your problem if you can convert the HBOOK to ROOT file (h2root)

Understanding RooRealVar

- All observables and parameters are of the type RooRealVar

bwg36.kek.jp - PuTTY

```
// Fit observables
RooRealVar Dam("Roo_Dam", "Recovered D* mass", 2.004, 2.02, " GeV/c^{2}");
RooRealVar Fksm("Roo_Fksm", "Ks mass", 0.0, 5.0, " GeV/c^{2}");
RooRealVar Fd0m("Roo_Fd0m", "D0 mass", 0.0, 10.0, " GeV/c^{2}");
RooRealVar Fdam("Roo_Fdam", "D* mass", 0.0, 10.0, " GeV/c^{2}");
RooDataSet* data = new RooDataSet("data", "data",RooArgSet(Dam,Fksm,Fd0m,Fdam));
```

  - Roo_Dam is the RooFit's internal identifier string for the observable Dam
  - The second argument is the description of the observable
  - The given range of the observable 2.004 and 2.02 (can put any values here, they will make sense only if they are within the observable original range)
  - The unit argument (given at the end) is optional

- Variable fixed at some value

```
RooRealVar frac("frac","Signal Core Fraction",0.52);
```

- Variable to be varied between two values, with an initial value

```
RooRealVar mean("mean","D^{*} mass",2.0099,2.0095,2.0103);
```

Getting a RooDataSet

- We need to create a RooDataSet object to hold the events we want to fit and plot thereafter

bwg36.kek.jp - PuTTY

```
Float_t x_Dam;
Float_t x_Fksm;
Float_t x_Fd0m;
Float_t x_Fdam;
// Read data file
TFile* input=new TFile("mydsks-on.root");
TTree* tree_data = (TTree*) input->Get("h1");
// no of entries in the datafile
Int_t n_tot = (int)tree_data->GetEntries();
cout << " n_tot = " << n_tot << endl;
Int_t n_sel(0);
tree_data->SetBranchAddress("Dam",  &x_Dam);
tree_data->SetBranchAddress("Fksm", &x_Fksm);
tree_data->SetBranchAddress("Fd0m", &x_Fd0m);
tree_data->SetBranchAddress("Fdam", &x_Fdam);
// Import the data points
for(int i=0; i<n_tot; i++) {
    tree_data->GetEntry(i);
    // fit region event selection
    //if(x_Dam>2.004&&x_Dam<2.02) {
    if (x_Dam>2.004&&x_Dam<2.02&&
        x_Fksm>0.0&&x_Fksm<3.0&&
        x_Fd0m>0.5&&x_Fd0m<8.0&&
        x_Fdam>0.5&&x_Fdam<8.0) {
      Dam.setVal(x_Dam);
      Fksm.setVal(x_Fksm);
      Fd0m.setVal(x_Fd0m);
      Fdam.setVal(x_Fdam);
      data->add(RooArgSet(Dam,Fksm,Fd0m,Fdam));
      n_sel++;
    }
                                                       13,3          19%
```

- Can be created from an ascii file or from a ROOT tree; I shall cover the 2nd way

Applying cuts

- Well, we have already applied the cut in a more traditional way, but RooFit also provides a framework for this

bwg36.kek.jp - PuTTY

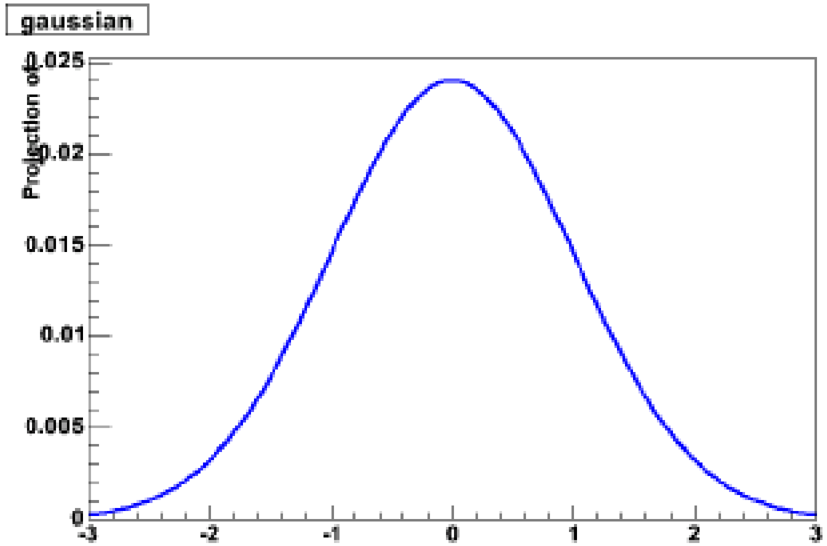
```
TCut myCut = "Dam>2.004&&Dam<2.02&&Fksm>0&&Fksm<3&&Fd0m>0.5&&Fd0m<8&&Fdam>0.5&&Fdam<8";
data->add(RooArgSet(Dam,Fksm,Fd0m,Fdam));
RooDataSet* reduce data = (RooDataSet*) data->reduce(myCut);
```

- Points to remember
  - Before you apply cut on some variable in RooFit way the variable must be in the RooDataSet
  - And, if you wish to cut on many variables this method is not useful since RooDataSet can not handle more than a finite number (I think 10)
  - I always prefer to reduce the main dataset putting all cuts before jumping to RooFit (anyway you don't want to use RooFit to apply cut)

Defining a PDF: RooGaussian

- Mostly used for the signal peak or, the peaking background

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left\{\frac{-(x-\mu)^2}{2\sigma^2}\right\}$$



- How did I do it?

bwg36.kek.jp - PuTTY

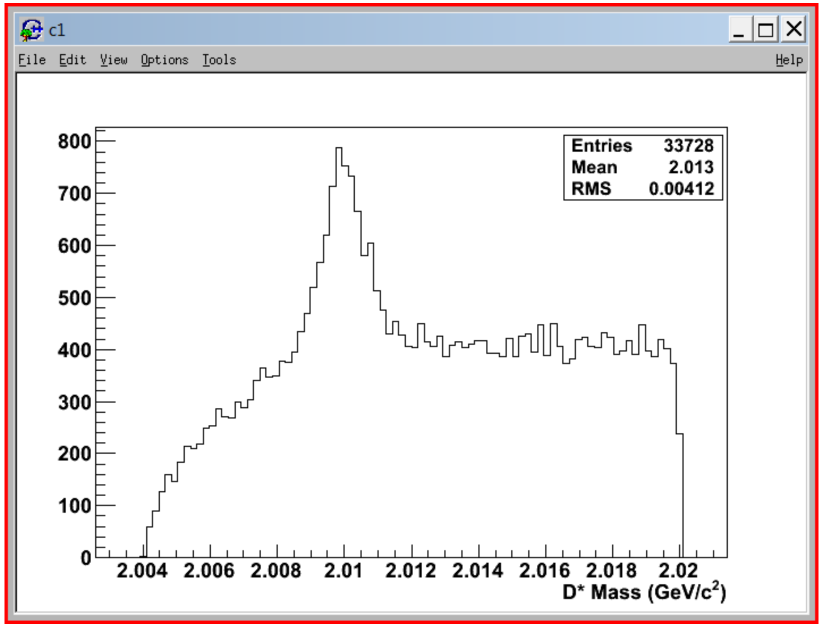
```
// Signal PDF
RooRealVar mean("mean","D^{*} mass",2.0099,2.0095,2.0103);
RooRealVar width1("width1","D^{*} width1",0.0004,0.0,0.001);
RooGaussian sig_core("sig_core","Signal Core Gaussian",Dam,mean,width1);
```

Can I have my own one?

- Yes, use RooGenericPDF

- Readily available threshold function (BN# 621) to fit the combinatorial background:

$$\sum_{i=1\ldots3} A_i \left[m-(m_D+m\pi)\right]^{\frac{2i-1}{2}}$$



bwg36.kek.jp - PuTTY

```
  // Background PDF
  RooRealVar a("a", "a", 0.04, 0.0, 1.0);
  RooRealVar b("b", "b", -1.3, -3.0, 3.0);
  RooRealVar c("c", "c", 14.9, -20.0, 40.0);
  RooGenericPdf bkg("bkg","a*sqrt(Dam-2.004)+b*(Dam-2.004)*sqrt(Dam-2.004)+c*(Dam-2.0
04)*(Dam-2.004)*sqrt(Dam-2.004)",RooArgSet(Dam,a,b,c));
  RooRealVar nBkg("nBkg", "nBkg",314300.0,1e4,12e5);
```

Adding PDFs together

➢ Now we need to add signal and background PDFs together to create the total PDF that we'll fit to the data

bwg36.kek.jp - PuTTY

```
  // Signal PDF
  RooRealVar mean("mean","D^{*} mass",2.0099,2.0095,2.0103);
  RooRealVar width1("width1","D^{*} width1",0.0004,0.0,0.001);
  RooGaussian sig_core("sig_core","Signal Core Gaussian",Dam,mean,width1);
  RooRealVar width2("width2","D^{*} width2",0.0012,0.0,0.005);
  RooGaussian sig_tail("sig_tail","Signal Tail Gaussian",Dam,mean,width2);
  RooRealVar frac("frac","Signal Core Fraction",0.52,0.0,1.0);
  RooAddPdf sig("sig","Signal PDF",RooArgList(sig_core,sig_tail),frac);
  RooRealVar nSig("nSig", "nSig",31400.0,1e3,12e5);

  // Background PDF
  RooRealVar a("a", "a", 0.04, 0.0, 1.0);
  RooRealVar b("b", "b", -1.3, -3.0, 3.0);
  RooRealVar c("c", "c", 14.9, -20.0, 40.0);
  RooGenericPdf bkg("bkg","a*sqrt(Dam-2.004)+b*(Dam-2.004)*sqrt(Dam-2.004)+c*(Dam-2.0
04)*(Dam-2.004)*sqrt(Dam-2.004)",RooArgSet(Dam,a,b,c));
  RooRealVar nBkg("nBkg", "nBkg",314300.0,1e4,12e5);

  RooAddPdf model("model","sig+bkg",RooArgList(sig, bkg), RooArgList(nSig, nBkg));
```

Now fitting to the data

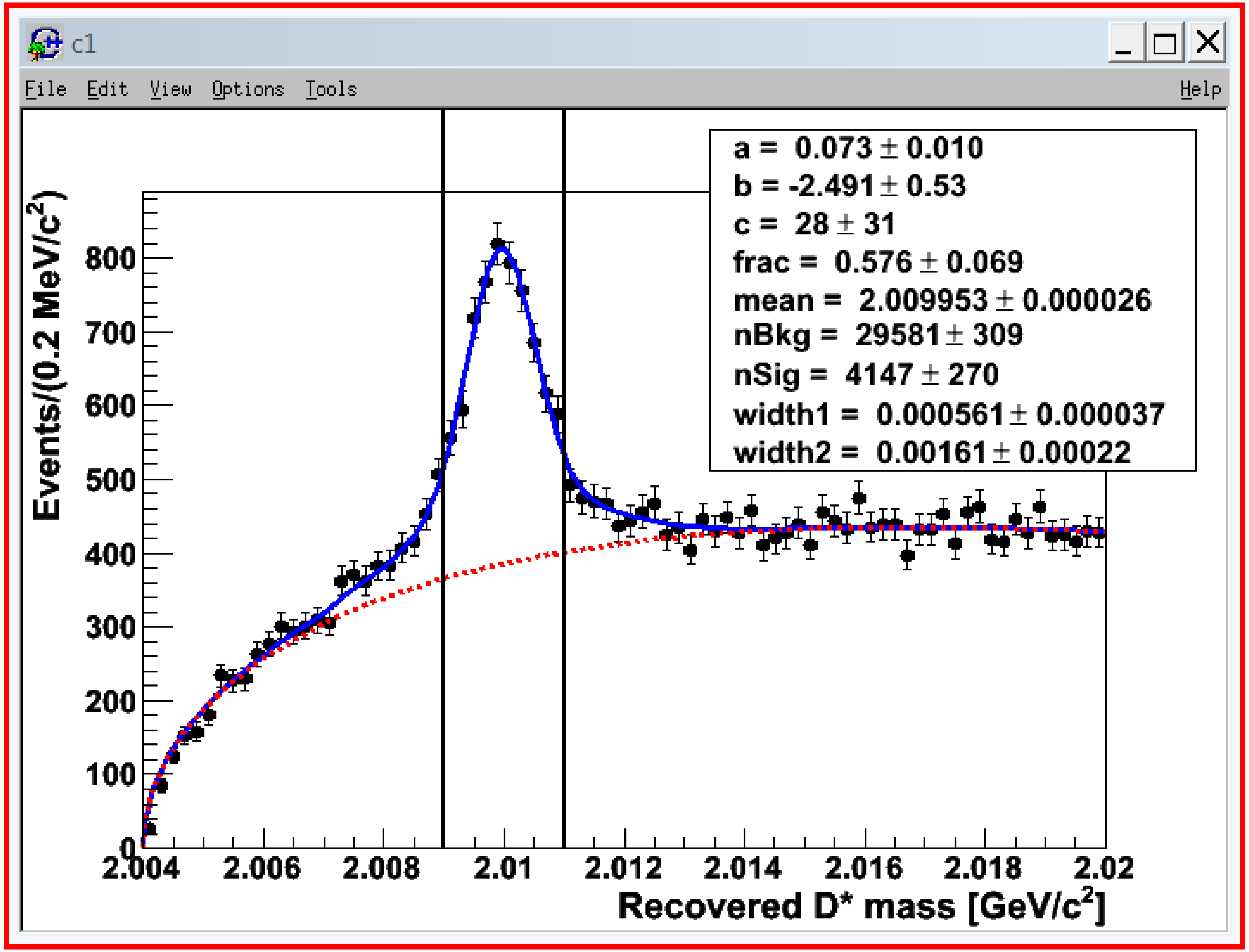
- For doing the fit, we just use:

```
RooFitResult* fitRes = model.fitTo(*data,RooFit::Minos(kFALSE),RooFit::Save(kTRUE));
```

  - I have switched off Minos ➡ No asymmetric error
  - Not yet doing an "extended" unbinned fit ➡ the normalization is fixed to the number of events in the histogram
  - You can use the extended option by having Extended() as the one of the arguments after *data
  - Save() as an argument to fitTo() is creating a RooFitResult object that would store many fit related information, such as covariance matrix

- Where is the result? Hold you breath

Here you go



c1
File Edit View Options Tools Help
a = 0.073 ± 0.010
b = -2.491 ± 0.53
c = 28 ± 31
frac = 0.576 ± 0.069
mean = 2.009953 ± 0.000026
nBkg = 29581 ± 309
nSig = 4147 ± 270
width1 = 0.000561 ± 0.000037
width2 = 0.00161 ± 0.00022
Events/(0.2 MeV/c²)
800
700
600
500
400
300
200
100
0
2.004
2.006
2.008
2.01
2.012
2.014
2.016
2.018
2.02
Recovered D* mass [GeV/c²]

Was it a good fit?

```
START MIGRAD MINIMIZATION.  STRATEGY  1.  CONVERGENCE WHEN EDM .LT. 1.00e-03
FCN=-226041 FROM MIGRAD    STATUS=INITIATE       40 CALLS          41 TOTAL
                     EDM= unknown      STRATEGY= 1      NO ERROR MATRIX
 EXT PARAMETER               CURRENT GUESS       STEP         FIRST
 NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
  1  a           4.00000e-02   2.00000e-02   1.05570e-01   9.98008e+02
  2  b          -1.30000e+00   6.00000e-01   2.25174e-01   5.29828e+02
  3  c           1.49000e+01   6.00000e+00   2.04259e-01   1.01090e+02
  4  frac        5.20000e-01   1.00000e-01   2.01528e-01   1.44670e+01
  5  mean        2.00990e+00   8.00000e-05   2.01358e-01  -3.13898e+01
  6  nBkg        3.14300e+05   1.19000e+05   2.33391e-01   4.68778e+05
  7  nSig        3.14000e+04   1.52000e+04   8.34425e-02   1.69064e+05
  8  width1      4.00000e-04   1.00000e-04   2.05758e-01  -8.32311e+01
  9  width2      1.20000e-03   5.00000e-04   2.39036e-01   6.95892e+01
                              ERR DEF= 0.5
MIGRAD FAILS TO FIND IMPROVEMENT
MACHINE ACCURACY LIMITS FURTHER IMPROVEMENT.
MIGRAD MINIMIZATION HAS CONVERGED.
MIGRAD WILL VERIFY CONVERGENCE AND ERROR MATRIX.
EIGENVALUES OF SECOND-DERIVATIVE MATRIX:
       -2.7080e-02  5.5620e-02  7.3350e-02  1.1449e-01  9.1064e-01  9.9184e-01  1.2656e+00  2.5013e+00  3.1142e+00
MINUIT WARNING IN HESSE
============== MATRIX FORCED POS-DEF BY ADDING 0.030194 TO DIAGONAL.
FCN=-459565 FROM HESSE     STATUS=NOT POSDEF     89 CALLS         783 TOTAL
                     EDM=0.000693482     STRATEGY= 1      ERR MATRIX NOT POS-DEF
```

- Seems not(!)
- Probably need to tweak some of the input parameters

```
MIGRAD FAILS TO FIND IMPROVEMENT
MIGRAD TERMINATED WITHOUT CONVERGENCE.
FCN=-459565 FROM MIGRAD    STATUS=FAILED        860 CALLS         861 TOTAL
                     EDM=14.773    STRATEGY= 1      ERR MATRIX NOT POS-DEF
 EXT PARAMETER                APPROXIMATE        STEP         FIRST
 NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
  1  a           7.28161e-02   7.43410e-01   0.00000e+00  -8.36545e-01
  2  b          -2.49104e+00   4.28698e+00  -0.00000e+00   1.17194e-01
  3  c           2.76792e+01   1.57450e+01  -0.00000e+00   8.22786e-02
  4  frac        5.76389e-01   7.55736e-01   0.00000e+00  -2.50286e-02
  5  mean        2.00995e+00   5.98336e-04   0.00000e+00  -1.25961e-02
  6  nBkg        2.95814e+04   9.26177e+05  -0.00000e+00  -1.48621e-01
  7  nSig        4.14658e+03   9.07427e+05  -0.00000e+00   6.45443e-01
  8  width1      5.61396e-04   7.44038e-04   0.00000e+00  -1.27146e-01
  9  width2      1.61256e-03   2.71083e-03   0.00000e+00  -9.18122e-02
                               ERR DEF= 0.5
 EXTERNAL ERROR MATRIX.    NDIM=  25    NPAR=  9    ERR DEF=0.5
  3.378e-01 -1.100e-02 -5.505e-01  1.175e-03 -1.000e-08 -8.943e+03  3.094e+03 -1.626e-07 -8.354e-06
 -1.100e-02  1.340e+01 -4.199e+01  6.550e-02  1.311e-06 -1.164e+04  3.731e+03  1.928e-06 -4.766e-04
 -5.505e-01 -4.199e+01  2.937e+02  4.371e+00  7.538e-05  4.631e+04 -3.664e+04  1.673e-04 -3.102e-02
  1.175e-03  6.550e-02  4.371e+00  1.225e+00 -1.132e-07  1.733e+03 -5.787e+02  1.774e-06  5.237e-05
 -1.000e-08  1.311e-06  7.538e-05 -1.132e-07  7.900e-07 -1.116e-01  3.902e-02 -1.293e-11  7.244e-10
 -8.943e+03 -1.164e+04  4.631e+04  1.733e+03 -1.116e-01  4.886e+10  2.295e+10 -1.346e+00 -1.070e+01
  3.094e+03  3.731e+03 -3.664e+04 -5.787e+02  3.902e-02  2.295e+10  1.095e+10  4.658e-01  3.505e+00
 -1.626e-07  1.928e-06  1.673e-04  1.774e-06 -1.293e-11 -1.346e+00  4.658e-01  1.238e-06  2.797e-09
 -8.354e-06 -4.766e-04 -3.102e-02  5.237e-05  7.244e-10 -1.070e+01  3.505e+00  2.797e-09  2.711e-05
ERR MATRIX NOT POS-DEF
```

Homework for the day

- Copy the ntuple and macro files from
- Go through each and every line and see if they make any sense
- Hopefully, you would understand it; if not ask me
- Fine tune input values of some/all parameters and if you want you can fix some of them
- But I need a good fit from all of you