More on the 2016 IMD

Emrys Peets
Stanford University

Unblinded ε^2 Upper Limit Results (from collab meeting)

Candidate Background Model Functional Form

$$\mathcal{C} \cdot \left[\mathrm{Er}_1 \cdot (1-x)^{p[1]} \cdot e^{p[2]\cdot\log(x)} + \mathrm{Er}_2 \cdot q[1] \cdot (1-x)^{q[2]} \cdot (1+x)^{q[3]\cdot x}\right]$$

Global Normalization Constant

Once candidate function has been determined

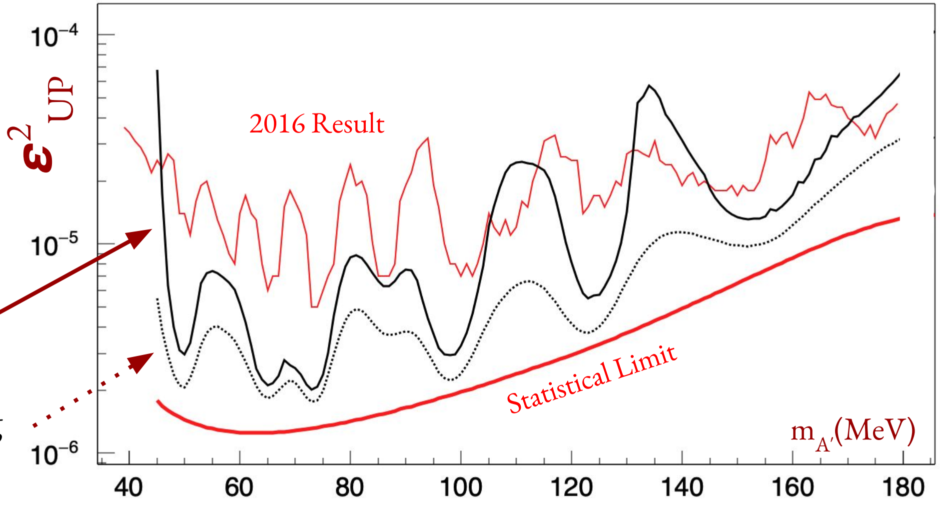
Fit over full IMD using HPS Analysis Software.

Study 1: all background parameters are floating
Study 2: only the global normalization constant is floating

Compute observed upper limits on signal yield and incorporate background + radiative fraction to determine ε^2 .

$$\epsilon^2 = \frac{2\alpha N_{\mathrm{sig}}^{\mathrm{up}}}{3\pi m_{A'} f_{\mathrm{rad}} \frac{\mathrm{d}N_{\mathrm{bkg}}}{\mathrm{d}m}}$$

ε^2 Upper Limit Comparison



Dilemma

Candidate Background Model Functional Form

$$\mathcal{C} \cdot \left[\mathrm{Er}_1 \cdot (1-x)^{p[1]} \cdot e^{p[2] \cdot \log(x)} + \mathrm{Er}_2 \cdot q[1] \cdot (1-x)^{q[2]} \cdot (1+x)^{q[3] \cdot x} \right]$$

Global Normalization Constant

Flawed..

This couples the shape characteristics of the functional form to the statistical properties (size / scaling).

Comparing Fitting with ROOT vs RooFit

ROOT

Pros

1. **Simplicity for Basic Cases:**
   a. For simple histograms (TH1) and functions (TF1), ROOT allows straightforward normalization by simply dividing the histogram by the total number of entries or scaling the function. This simplicity is suitable for basic cases where advanced normalization is not needed.
2. **User Control:**
   a. Users have complete control over the normalization process, which can be beneficial when a specific normalization method is required, or when integrating over non-standard ranges

Cons

1. **Manual Normalization:**
   a. ROOT does not automatically handle normalization for PDFs or fits. Users must manually ensure that the functions or histograms are properly normalized, which can be error-prone.
2. **Limited Flexibility:**
   a. ROOT's basic fitting tools do not provide the same level of flexibility in normalization as RooFit, particularly for composite models or models requiring integration over multiple variables.
3. **No Built-in Numerical Integration for PDFs:**
   a. ROOT's basic tools lack built-in support for numerical integration of PDFs, which can limit their utility for complex models that require accurate normalization over a range.

Comparing Fitting with ROOT vs RooFit

RooFit

Pros

1. **Automatic Normalization**
   a. RooFit automatically normalizes probability density functions (PDFs) when performing fits, ensuring that the integral of the PDF over the defined range is unity. This feature is particularly useful when dealing with composite models or multiple datasets.
2. **Flexible Normalization Options**
   a. It provides flexible options for normalization, allowing users to define custom normalization ranges or variables, which is useful for more complex or non-standard models.
3. **Efficient Integration**
   a. RooFit uses optimized numerical integration techniques for normalization, which can handle both analytical and numerical integration efficiently, even for complex or multi-dimensional PDFs.
4. **Consistency Across Fits**
   a. Because RooFit handles normalization internally and consistently, it reduces the risk of errors and ensures that different components of a model are normalized in a consistent way.

Cons

1. **Performance Overhead**
   a. The automatic and flexible normalization features can introduce additional computational overhead, especially when dealing with high-dimensional or complex models.
2. **Learning Curve**
   a. Understanding how to properly set up and control normalization in RooFit can add to the learning curve, especially for users unfamiliar with the framework.

Integrating RooFit into Global Fitting Infrastructure

SLAC

I. Tedious

Having to rewrite majority of fitting code, function txt files

II. Determines which parameters to fix/float based on its own fitting algorithm

III. Have achieved one good fit since starting to rework everything

RooFit Current Outputs

EIGENVALUES OF SECOND-DERIVATIVE MATRIX:

-1.3349e-03 7.6037e-05 1.2005e-03 2.5386e-03 7.0602e-03 2.7104e-01 5.0418e-01 7.2152e+00

EXT NO.	PARAMETER NAME	VALUE	APPROXIMATE ERROR	INTERNAL STEP SIZE	INTERNAL VALUE
1	[p10]	3.50185e+02	3.53317e+00	4.19254e-06	-5.62201e-01
2	[p1]	2.95819e-02	1.90605e-05	3.29714e-07	1.61103e-03
3	[p2]	8.14971e-02	3.11089e-05	1.86700e-07	-3.42042e-02
4	[p3]	7.15171e+01	3.20510e-01	2.59714e-06	3.44162e-01
5	[p4]	3.05917e+01	3.34937e-02	2.02217e-06	1.85304e-01
6	[p5]	-2.42447e+00	fixed		
7	[p6]	1.08256e-02	3.38710e-05	3.68695e-06	-2.11428e-02
8	[p7]	3.81382e-02	2.82470e-05	4.00985e-07	6.77836e-02
9	[p8]	-8.39900e+05	fixed		
10	[p9]	1.70768e+02	1.22582e-01	3.51350e-07	5.77961e-02

ERR DEF= 0.5

EXTERNAL ERROR MATRIX. NDIM= 25 NPAR= 8 ERR DEF=0.5

1.248e+01	2.400e-05	-1.523e-05	2.572e-01	-1.279e-02	4.646e-05	-6.970e-08	1.711e-01
2.400e-05	3.633e-10	2.474e-10	-5.783e-07	-1.424e-08	1.281e-10	9.783e-12	-9.063e-08
-1.523e-05	2.474e-10	9.678e-10	1.982e-06	7.042e-07	-4.361e-11	8.553e-11	1.007e-06
2.572e-01	-5.783e-07	1.982e-06	1.027e-01	2.950e-03	-5.027e-07	-3.089e-06	-6.659e-03
-1.279e-02	-1.424e-08	7.042e-07	2.950e-03	1.122e-03	1.468e-08	2.509e-07	1.056e-03
4.646e-05	1.281e-10	-4.361e-11	-5.027e-07	1.468e-08	1.147e-09	7.277e-10	-5.774e-07
-6.970e-08	9.783e-12	8.553e-11	-3.089e-06	2.509e-07	7.277e-10	7.979e-10	-5.477e-07
1.711e-01	-9.063e-08	1.007e-06	-6.659e-03	1.056e-03	-5.774e-07	-5.477e-07	1.503e-02

Fit 10: Chi2 = 2643.78624600157, Chi2 Probability = 0.317447903358967 44
Best fit: Chi2 = 2642.581200034163, Chi2 Probability = 0.323365392638887 97

PARAMETER CORRELATION COEFFICIENTS

NO.	GLOBAL	1	2	3	4	5	7	8	10
1	0.99426	1.000	0.356	-0.139	0.227	-0.108	0.388	-0.001	0.395
2	0.99347	0.356	1.000	0.417	-0.095	-0.022	0.198	0.018	-0.039
3	0.99382	-0.139	0.417	1.000	0.199	0.676	-0.041	0.097	0.264
4	0.99456	0.227	-0.095	0.199	1.000	0.275	-0.046	-0.341	-0.169
5	0.92859	-0.108	-0.022	0.676	0.275	1.000	0.013	0.265	0.257
7	0.96403	0.388	0.198	-0.041	-0.046	0.013	1.000	0.761	-0.139
8	0.99317	-0.001	0.018	0.097	-0.341	0.265	0.761	1.000	-0.158
10	0.99437	0.395	-0.039	0.264	-0.169	0.257	-0.139	-0.158	1.000

Nest Steps

I. Finish reworking displays to incorporate RooFit
II. Once completed for 6.5%, rewrite parameter projection code to see if it works for 100%
III. Test on histograms being made with signal injected
IV. rewrite bumphunter to take into account expected values using roofit