Global Fitting Update

Emrys Peets - Stanford University
Cameron Bravo - SLAC
February 6, 2024

U.S. DEPARTMENT OF ENERGY
Office of Science
SLAC NATIONAL ACCELERATOR LABORATORY

Update Since the Collaboration Meeting

I. Implemented global fitting methods into hpstr

II. Successful in testing top performing function from previous function selection

III. Successful in throwing toy distributions, but have not run with full stats

IV. Plotted signal yield upper limits and epsilon2 upper limits for 2016 data with background floating in bkg only fit and :
   A. background floating in bkg+signal fit
   B. background fixed in bkg+signal fit
   C. modified background function with normalization constant
      1. normalization constant floating, all other parameters fixed in bkg+signal fit

V. Note has been started and rough draft is on confluence/overleaf
   A. https://confluence.slac.stanford.edu/display/hpsg/Physics+Analysis+Notes
   B. https://www.overleaf.com/read/qfbmpfbwfrzn#87bb7

Changes in hpstr

Modified:

/hpstr/analysis/src(include)/BumpHunter.cxx (.h)

/hpstr/analysis/src(include)/las3pluslas6_FitFunction.cxx(.h)

/hpstr/analysis/src(include)/HpsFitResult.cxx (.h)

/hpstr/analysis/src(include)/FunctionMath.cxx(.h)

/hpstr/analysis/include/FitFunction.h

/hpstr/processors/src/BhToysHistoProcessor.cxx

/hpstr/analysis/config/bhToys_cfg.py

Added:

/emrys_files/makeBhToysStudyScripts.py

(original function)

```
double FunctionMath::las3pluslas6_Function(double x, double* p) {

    return ((TMath::Erf((x-p[1])/p[0])+1)/2 * p[2]*TMath::Power((1-x),p[3])*TMath::Exp(p[4]*log(x)))
+((TMath::Erf((x-p[6])/p[5])+1)/2 * p[7]*TMath::Power((1-x),p[8])*TMath::Power(1+x,p[9]*x));

}
```

las3pluslas6 Function modification

Function has been modified to incorporate a global normalization constant, where the starting parameters for c2 are changed to equal (c2/c1).

$$\text{las3pluslas6}(x) = \left(c_1 \cdot \text{las3}'(x)\right) + \left(c_2 \cdot \text{las6}'(x)\right)$$

$$\rightarrow = c_1 \cdot \left(\text{las3}' + \frac{c_2}{c_1} \cdot \text{las6}'\right)$$

function mod

```
    return p[2]*(((TMath::Erf((x-p[1])/p[0])+1)/2*TMath::Power((1-x),p[3])*TMath::Exp(p[4]*log(x)))
+((TMath::Erf((x-p[6])/p[5])+1)/2*p[7]*TMath::Power((1-x),p[8])*TMath::Power(1+x,p[9]*x)));
```

starting parameter mod →

```
bkg->SetParameter(0, 0.02655677447001521);
bkg->SetParameter(1, 0.09575583442743552);
bkg->SetParameter(2, 1.6087608867103269e-06);
bkg->SetParameter(3, -12.14155381679078);
bkg->SetParameter(4, -9.88122176150782);
bkg->SetParameter(5, -0.015730267362833915);
bkg->SetParameter(6, 0.11327528231496534);
bkg->SetParameter(7, -14701589.955451723 / 1.6087608867103269e-06);
bkg->SetParameter(8, 117.94823473423622);
bkg->SetParameter(9, 423.73510122988904);
```

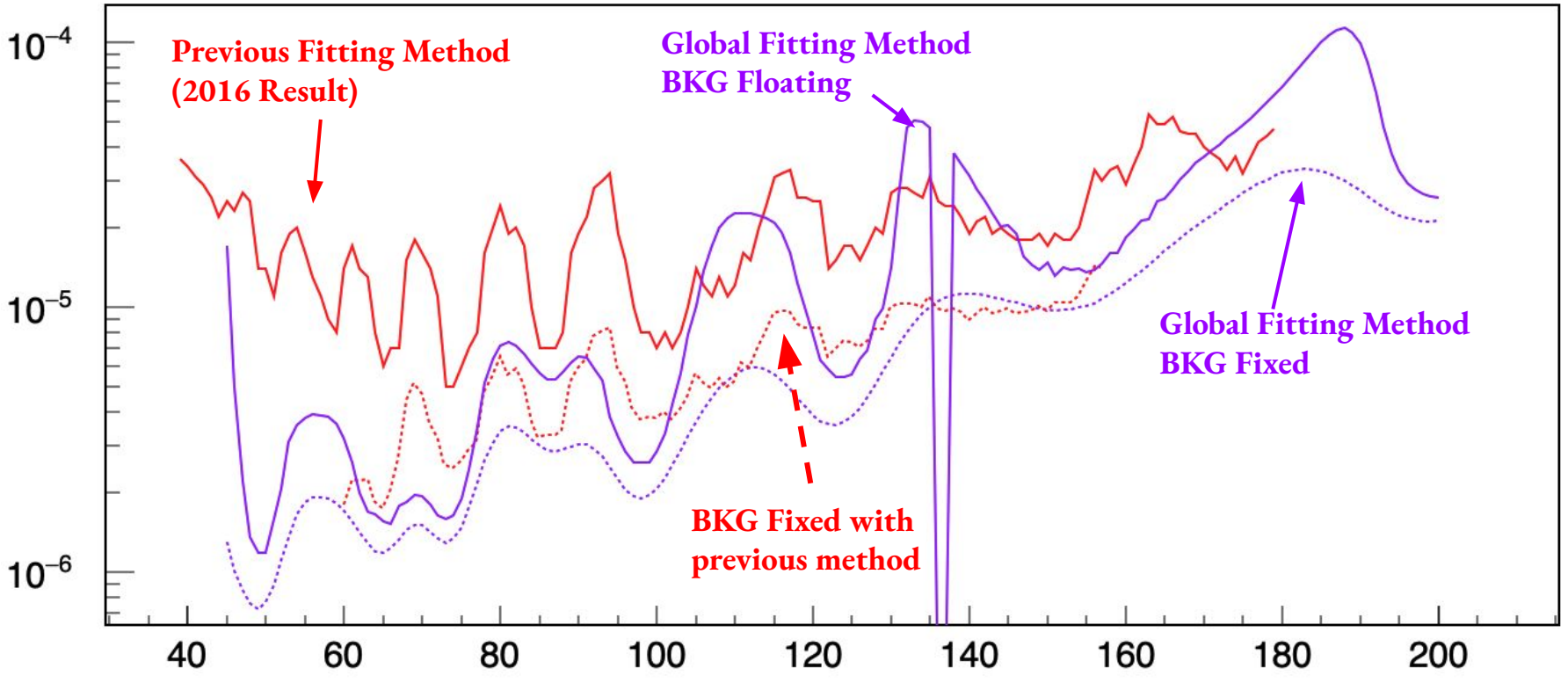
Fitting procedure for each mass hypothesis

I. Set fit range to [45, 200] MeV over the IMD
   A. Determined in function selection testing
   B. las3pluslas6 bkg fit had a chi2 probability of 5.8e-2 for that range

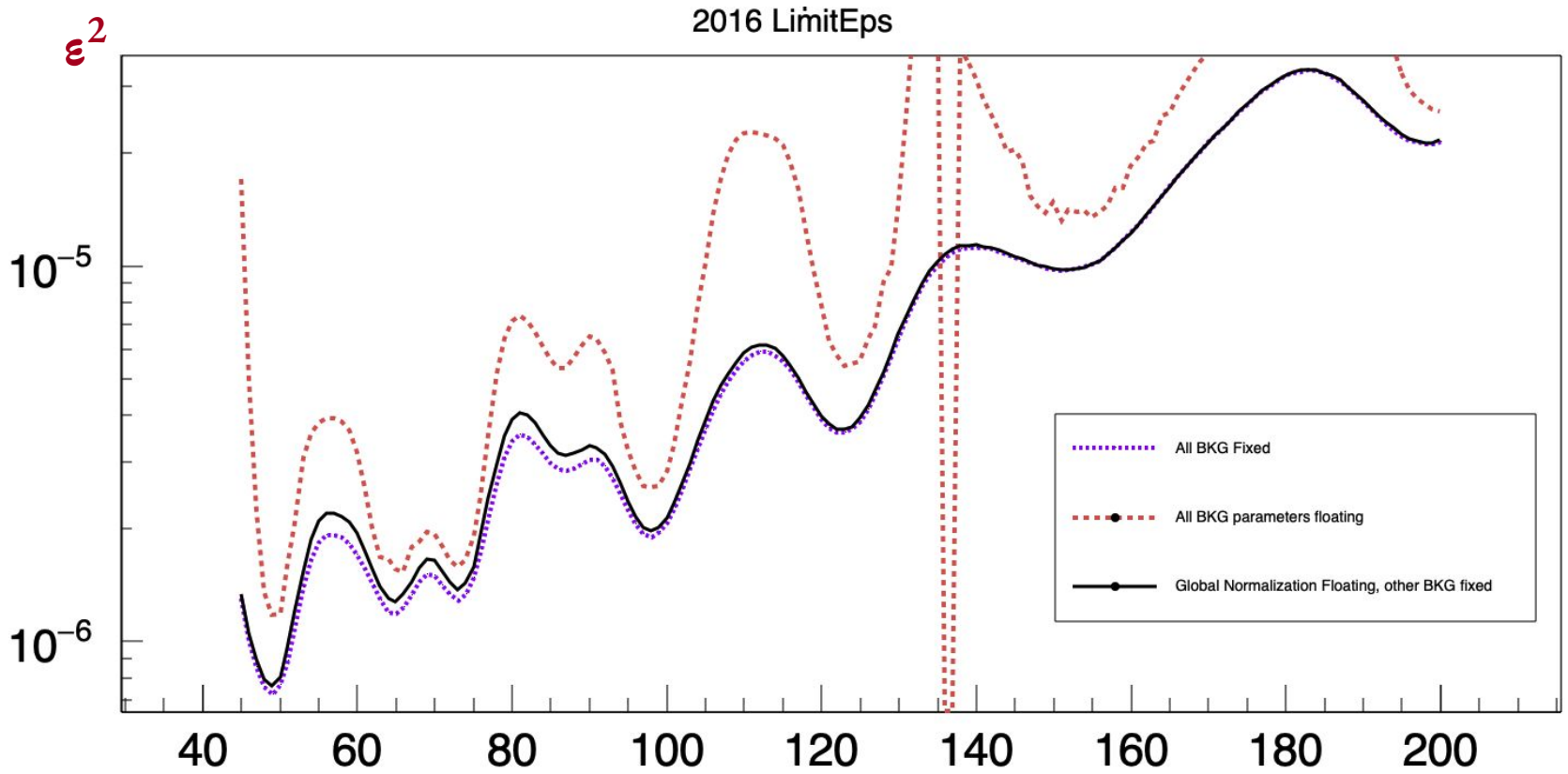
II. Background Only fitting
   A. Set parameters to those found from best fit over interval from ^^

III. Background + Signal Fitting
   A. create full function using signal gaussian
      a) signal normalization floating
      b) mass hypothesis, mass resolution fixed
   B. three different background models tested
      1. all bkg parameters floating
      2. all bkg parameters fixed
      3. modified function with global normalization constant floating and all other bkg parameters fixed

ε^2 Upper Limit Comparison



Comparison of Global Fitting background models' corresponding ε^2 upper limits



Next Steps

- signal injection study
- continue writing note
- run with toys to create limit bands