Hyper-EMG User Manual

A basic understanding of how the “Hyper-EMG” and “Find Fit Uncertainty” codes functions can be found in the “code roadmap” file availably in the Titan GitHub repository. This document will focus on the understanding what needs to be done to effectively use the code.

Table of Contents

[1. Exporting .txt files from the MAC software 2](#_Toc31299335)

[2. Code: Hyper-EMG.py 2](#_Toc31299336)

[2.1. PLEASE READ: IMPORTANT INFORMATION 2](#_Toc31299337)

[2.2. Reference Image 2](#_Toc31299338)

[2.3. Getting the Code to Work for Your Data 3](#_Toc31299339)

[2.3.1. Step 1 – Inserting file paths (lines 1110 – 1112): 3](#_Toc31299340)

[2.3.2. Step 2 – Identifying peak centres (lines 1114 – 1119): 4](#_Toc31299341)

[2.3.3. Step 3 – Entering number of tails (lines 1121 – 1122): 4](#_Toc31299342)

[2.3.4. Step 4 – Picking your initial guesses (lines 1124 – 1134): 4](#_Toc31299343)

[2.3.5. Step 5 – Functions responsible for graphing (lines 1152+): 4](#_Toc31299344)

[2.4. Explaining the User Inputted Initial Guesses (lines 1124 – 1134) 4](#_Toc31299345)

[2.4.1. Exponential tails (tp1, np1…) 5](#_Toc31299346)

[2.4.2. Final weighting factor (weight) 5](#_Toc31299347)

[2.4.3. Gaussian core 5](#_Toc31299348)

[2.5. Displaying fit data for exporting (lines 1146-1148): 6](#_Toc31299349)

[2.6. Understanding the Outputs 6](#_Toc31299350)

[2.6.1. Calibration file fit results 6](#_Toc31299351)

[2.6.2. Doublet/triplet peak fit results 7](#_Toc31299352)

[2.7. Optimizing Your Fit 7](#_Toc31299353)

[2.7.1. Increasing or decreasing the number of tails 7](#_Toc31299354)

[2.7.2. Optimizing your initial guesses 8](#_Toc31299355)

[3. Code: Find Fit Uncertainty.py 9](#_Toc31299356)

[3.1 Reference Image 9](#_Toc31299357)

[3.2. Explaining the Uncertainties 10](#_Toc31299358)

[3.2.1. Peak shape uncertainty 10](#_Toc31299359)

[3.2.2. Statistical uncertainty 10](#_Toc31299360)

[3.3 Transferring Relevant Information from Previous Code 10](#_Toc31299361)

[3.3.1. Selecting the number of tails used (lines 426- 427) 10](#_Toc31299362)

[3.3.2. Copying over the file paths (lines 445-449) 10](#_Toc31299363)

[3.3.3. Copying over the optimized parameter values (lines 449 – 447) 10](#_Toc31299364)

[3.3.4. Plotting (lines 549,554,559) 11](#_Toc31299365)

[3.4. Interpreting Results 11](#_Toc31299366)

# 1. Exporting .txt files from the MAC software

Important notes:

* The .txt files need to be exported from the MAC software in histogram form.
* Headers within the exported .txt files will need to be erased. This is done manually by opening the .txt file and removing all rows not containing data.
* **Try to export only the close proximity to the peaks of interest. Extra and excessive 0 value bins to the left or right of your peak can result in an artificially lower x2.**

You will be required to export two separate .txt files for your data:

1. The “calibrant peak” file:

* This file should contain one, well isolated peak, denoted as the “peak shape calibrant”. This is the peak that the H-EMG peak shape will be based off of.

1. The “peak of interest” file(s):

* These files should contain at maximum 3 peaks. These are the peaks that the H-EMG peak shape will be superimposed onto.

# 2. Code: Hyper-EMG.py

## 2.1. PLEASE READ: IMPORTANT INFORMATION

The found absolute center value for the peaks that are fit will not be consistent with known values. Instead, you must only **use the difference between two peaks**. One peak needs to correspond to a known mass value, the other corresponding to the ion of interest.

## 2.2. Reference Image

All user inputted values can be found on lines 1110-1153 within the python file. The lines referenced in this document refer to the lines as displayed in the following screenshot:

A screenshot of text

Description automatically generated

## 2.3. Getting the Code to Work for Your Data

The following steps outline the actions that the user would need to take to run the code.

### 2.3.1. Step 1 – Inserting file paths (lines 1110 – 1112):

First you will need to let the code know where to find the data files you have exported. This can be done on lines 1109 through 1111 of the code. You will need to enter the entire file path:

* ‘calibfile’ refers to your peak shape calibrant file.
* ‘triplePeakFile’ refers to a data file containing three peaks to be fit.
* ‘doublePeakFile’ refers to a data file containing two peaks to be fit.

Should you not have need for fitting a double peak / triple peak, you can leave the corresponding string blank, **but do not delete the variable or the quotation marks associated with the variable.**

### 2.3.2. Step 2 – Identifying peak centres (lines 1114 – 1119):

Insert where you believe the centroid of each peak should be, in AMU. The curve fitting function will need an initial guess for where to find the centroid of each peak (in AMU). These inputs can be crude approximations of where you believe the peak to be. As long as the guesses are reasonable, the final centroid will not be dependent on the initial guesses.

### 2.3.3. Step 3 – Entering number of tails (lines 1121 – 1122):

Choose the number of negative (left side of peak) and positive (right side of peak) exponential tails you wish to use when fitting the calibration peak. The maximum number of exponential tails is three for each side. More detail on how to choose the number of tails you may need to model your peak is explained in the “Optimizing your fit” section.

### 2.3.4. Step 4 – Picking your initial guesses (lines 1124 – 1134):

Choosing the initial guesses is a pseudo random process at first. You can somewhat guess these values depending on your peak shape (elongated left/ right tails). Mainly, peak fitting result of your first or first few runs will point you towards the right direction for future optimization.

To understand what these variables are referring to, see the “Explaining the User Inputted Initial Guesses”.

### 2.3.5. Step 5 – Functions responsible for graphing (lines 1152+):

These lines contain the functions that actually call the rest of the code into action:

* The “﻿FitTriplePeak” function is responsible for fitting the found peak shape to a data set with 3 peaks.
* The “﻿FitDoublePeak” function is responsible for fitting the found peak shape to a data set with 2 peaks.

If you only have need to fit to a data set containing three peaks, or two peaks respectively, you can simply comment out (# at the front of the line) or erase the function pertaining to the fit type you will not be utilizing.

#### Fitting multiple triplet/doublet data files

More advanced users could duplicate these lines of code to call the plotting function again on a separate data set. You would then need to create and assign new variables to be used (path, initial centroid guesses…) in this copied function.

## 2.4. Explaining the User Inputted Initial Guesses (lines 1124 – 1134)

A hyper exponentially modified gaussian consists of a gaussian center, with a variable number of exponential tails convoluted into the positive and negative tail of the gaussian. This code fits a Hyper-EMG to the peaks of interest through the least square optimization of the various free parameters associated with the exponential tails and Gaussian core. You will require a basic understanding of these free parameters to efficiently optimize your fit.

### 2.4.1. Exponential tails (tp1, np1…)

Each exponential tail contains its own set of free parameters; a time constant tau and normalization factor eta.

In the code, these free parameters are…

A picture containing object, clock

Description automatically generated A picture containing object, clock

Description automatically generated

**tp1** = Time constant tau pertaining to the first positive (right side of gaussian) tail.

**np1** = Normalization eta factor pertaining to the first positive (right side of gaussian) tail.

**tm1** = Time constant tau pertaining to the first negative (left side ofcentre) tail.

**nm1** = Normalization eta factor pertaining to the first negative (left side of gaussian) tail.

And so on.

Notes:

* The sum of all eta for the positive and negative sides seperately are equal to 1. This means while there exists a time contant for every tail, there exists (# of tails – 1) eta perameters.
* I.E. if you are fitting to 2 left tail, you will only need to alter tm1, tm2, and nm1, NOT nm2, since nm1+nm2=1.
* Eta can only take values between 0 and 1.

### 2.4.2. Final weighting factor (weight)

There is a normalization weighting factor for the left and right tails summed together.

A picture containing table, clock

Description automatically generated

This, in the code, corresponds to the initial parameter “**weight**”. Notice that a weight of 0.9 would heavily favor the negative tail of the Hyper-EMG, while a weight of 0.1 would favor the positive side.

### 2.4.3. Gaussian core

The core gaussian can be characterized by three free parameters: The centre, width, and amplitude.

A picture containing object, clock

Description automatically generated

Note that the initial guess for these variables are taken from a gaussian fit to the calibrant peak, and as such do not require the user to input an initial guess.

## 2.5. Displaying fit data for exporting (lines 1146-1148):

These lines of code export the specific fit for a designated peak. You can choose the specific peak to print using the following syntax:

* True/False index:
* True = You wish to print out the fit data for one peak/all peaks in that dataset.
* False = You do not wish to print any fit data for the peaks in that dataset.
* Numeric index:
* 0 = The sum peak.
* 1 = Farthest left individual peak.
* 2 = Second peak from the left.
* 3 = Third peak from the left.

Example:

If I wish to print only the fit information for the middle peak of a triplet data set, lines 1146 through 1148 would appear as follows:

﻿displayCalibFitData=[False,0]

displayTripletFitData=[True,2]

displayDoubletFitData=[False,0]

## 2.6. Understanding the Outputs

### 2.6.1. Calibration file fit results

#### Information on the data

* The total number of counts within the calibration peak.
* The integral of the now normalized calibration peak; should be approximately 1.
* The standard deviation and centroid of a standard gaussian’s fit on the calibrant data.

#### Information about the resulting calibrant Hyper-EMG fit

* The optimized free parameter values based on the inputted initial guesses. The first column is the optimized value itself, and the second being the uncertainty on this value.
* The scaling factor is fixed to 1 since we fit to normalized data.
* The area under the curve of the fitted Hyper-EMG; should be approximately 1.
* The resulting chi squared for the Hyper-EGM fit on the data.

#### Graph 1- the data and fit

* Green - A simple gaussian that has been fit to the calibration peak.
* Blue - The fitted Hyper-EMG.
* Black - The data.
* The uncertainty on the data is not displayed.
* This is a semi-log graph.

#### Graph 2- The residuals

* This is a graph of the data, minus the fit. It shows where the function overestimated or underestimated the shape of the peak.

### 2.6.2. Doublet/triplet peak fit results

#### Information about the resulting Hyper-EMG fit

The optimized free parameter values. Only the amplitude and centroids are optimized:

* AIx/AIIx/AIIIx - Amplitude and its uncertainty for the left-most peak, second and third from left peaks.
* CIx/CIIx/CIIIx - Centroid and its uncertainty for the left-most peak, second and third from left peaks.
* Peak separation (b-a) - The separation between peaks a and b, where a and b correspond to the left-most (1), second from left (2), and the third from left (3) peaks.

Resulting counts in data:

* The total number of counts in the data. This is directly based off the data.
* The rounded integral of the left-most, second and third from left peaks.

#### Graph 1- the data and fit

* Blue - The sum fit of the Hyper-EMG on all peaks.
* Red - The individual peak fits that make up the sum fit.
* Black - The data.
* The uncertainty on the data is not displayed.
* This is a semi-log graph.

#### Graph 2- The residuals

* This is a graph of the data, minus the fit. It shows where the function overestimated or underestimated the shape of the peak.

## 2.7. Optimizing Your Fit

There are several indicators that would tell you if you can optimize your fit further:

1. If your x2 is high, or much less then 1 for the peak shape calibrant.
2. If the uncertainty on your optimized parameters is larger than the parameter itself.
3. If the final uncertainty found using the “Find Fit Uncertainty” code is unreasonably large.
4. You are getting a bad fit.

### 2.7.1. Increasing or decreasing the number of tails

There is 1 main indicator for needing to **increase the number of tails:**

1. The fit is routinely underestimating a portion of the tail in the data.

There are 3 main indicators for needing to **decrease the number of tails**:

1. Your found x2 is smaller than 1:

* This indicates you are overfitting your data, decreasing the number of free parameters will help you better model the data.

1. One of the optimized **eta** values is very small (order around E-3 or smaller) **OR** the uncertainty on the found value is extremely large:

* Eta values are the weighting factors for that exponential tails. A low eta means that the contribution of that exponential tail is very low.
* A high uncertainty suggests that any value could be used for this eta. This implies that the tail has no significant effect on the fit.

1. One of the optimized **tau** parameters is very small (order of E-5 or smaller) **OR** the uncertainty on the found value is extremely large:

* An exponential tail with a very low time constant is essentially a vertical line. This means that the optimizer found this tail to not contribute significantly to the peak shape.
* A high uncertainty suggests that any value could be used for this eta. This implies that the tail has no significant effect on the fit.

### 2.7.2. Optimizing your initial guesses

These are tricks I adapted when trying to optimize my found parameter values. These optimization tricks might help reduce the found uncertainty on a fit.

#### Step 1 – Identify what is wrong with your fit

Look at your results. What is specifically wrong? Is the left tail being underestimated? Right tail considering the noise to much? These will give you clues on how to proceed.

#### Step 2 – Try: change the ‘weight’ initial guess

In my experience, this had the largest effect on the peak shape. It controls the weight given to all left or right-side exponential tails. If one side is not being fit well, while the other is perfectly fit, try to shift the weighting towards the side that is poorly fit.

#### Step 3 – Try: changing a tau or eta value

If you think the tail needs to be longer, try increasing one of the initial tau guesses. If it’s too long, try a smaller tau. Alternatively, try to shift around the eta weightings. Adding more emphasis on one tail (increasing its weighting) would have a similar effect to increasing the tau value of that tail.

#### Step 4 – Observe the changes

Repeat steps 2 and 3 as you see fit until you are satisfied with the fit. Don’t forget to try increasing or decreasing the number of tails.

#### Additional Info:

The fit is optimized using a least square fit, which essentially locates a ‘local minimum’ in its found value of x2 for each parameter. Given a different start point, the optimizer might find a different ‘local minimum’ for a given parameter. Thus changing your initial guesses will change your optimized parameter values, but should not significantly alter the found peak shape (if the found peak shape is a good fit!)

# 3. Code: Find Fit Uncertainty.py

## 3.1 Reference Image

All user inputted values can be found on lines 426 to 477 within the python file. The lines referenced in this document refer to the lines as displayed in the following screenshot:

A screenshot of text

Description automatically generated

## 3.2. Explaining the Uncertainties

There are 2 main uncertainties associated with a Hyper-EMG fit.

### 3.2.1. Peak shape uncertainty

The peak shape uncertainty is the uncertainty associated with the centroid value of a single peak due to the uncertainty in the optimized parameters. It is found by varying each parameter by its uncertainty and seeing how it effects the movement of the found centroid.

### 3.2.2. Statistical uncertainty

This is the uncertainty associated with the mean of the Hyper-EMG. It is proportional to the root of the number of counts within the peak.

## 3.3 Transferring Relevant Information from Previous Code

### 3.3.1. Selecting the number of tails used (lines 426- 427)

Ensure that lines 426 and 427 have the same number of tails as used when peak shape fitting.

### 3.3.2. Copying over the file paths (lines 445-449)

Ensure that the paths to the corresponding .txt files are correct.

Important:

Unlike before, if one of the doublet or triplet peaks were not used, the path variable corresponding to the unused function type will need to be filled with a placeholder. I recommend simply copying over one of the other two paths, then simply ignoring the pertinent results.

### 3.3.3. Copying over the optimized parameter values (lines 449 – 447)

The optimized parameters for each fitted peak needs to be copied over from the previous code. To do so, simply highlight the parameters, copy, and paste those values into the corresponding location. You can simply delete or replace any values that are in that location at the time.

Notice that you only need the amplitudes and centroids of the triplet/doublet peaks, not the separations.

Reminder:

Tab the values so they are in line with the rest of the code, or it will return an error.

Example:

A screenshot of text

Description automatically generated

Copy only the optimized free parameter values directly from the ‘Hyper-EMG’ code to the corresponding location in the ‘find fit uncertainty’ code, in this case, the “calibration” parameters are copied over.

### 3.3.4. Plotting (lines 549,554,559)

If the found uncertainties are extremely large, or you simply wish to see how the changes in the free parameters effect the fit, then remove the hashtag in front of any of the three lines listed above. After each parameter variation, a plot of the resulting fit will be printed.

* Line 549 corresponds to the calibrant data.
* Line 554 corresponds to the doublet data.
* Line 559 corresponds to the triplet data.

## 3.4. Interpreting Results

The displayed uncertainties are the absolute uncertainty for each individual peak in units of keV. If you did not need to find the doublet/triplet peak uncertainty calculation, simply ignore those results.

IMPORTANT:

Since you will be taking a difference between two peaks to determine the mass value of your ion of interest, you will have to propagate the error of both peaks together using the rules for uncertainty propagation in subtractions.