

# A three-color single-molecule FRET approach to studying correlated interactions in the heat shock protein 90

Markus Götz, Philipp Wortmann, Sonja Schmid, Thorsten Hugel  
Albert-Ludwigs-Universität Freiburg, Freiburg, Germany

## Software Kit User Manual

Version 1.1

## Installation & General Considerations

The scripts run with Igor Pro 6 ([https://www.wavemetrics.com/order/order\\_igordownloads6.htm](https://www.wavemetrics.com/order/order_igordownloads6.htm)). After downloading the source code and example data from <http://www.singlemolecule.uni-freiburg.de/software>, double click Main.ipf and everything is ready to run. In case the “3D HMM” menu does not show up automatically, click the compile button indicated in Fig. 1.

Elements in the user interfaces or menus are indicated by quotes. Commands that can be pasted to the Command Line (c.f. Fig. 1) are indicated by >>.

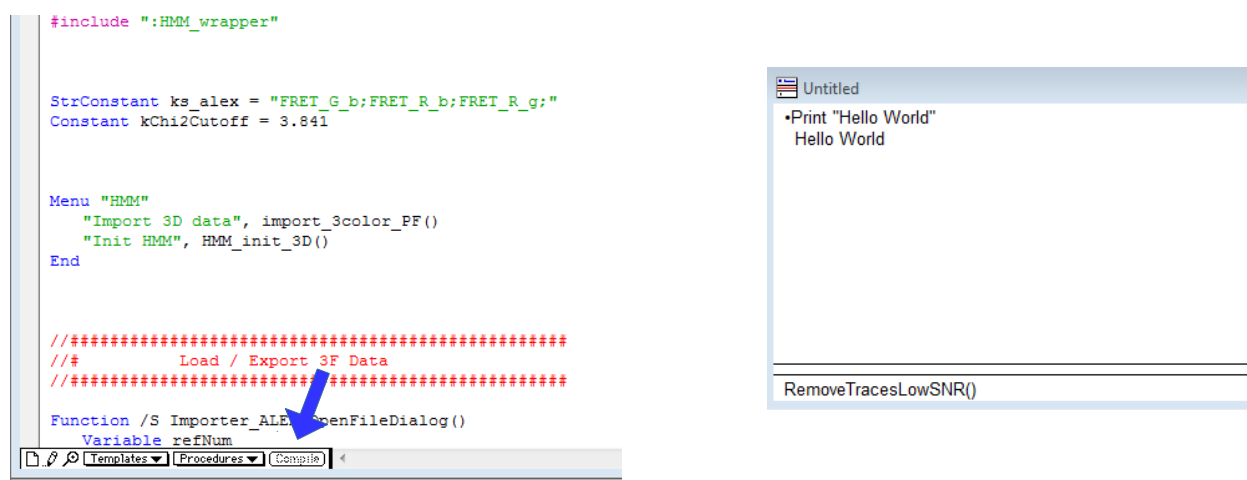


Figure 1: The "Compile" button (it vanishes after compilation) and the Command Line.

## Data Import

For maximum compatibility, the software kit comes with an ascii importer. In order to get an impression of the functionality, we further include experimental example data. To load the three-color smFRET data, go to the “3D HMM” menu → “Import 3D data”. The importer simultaneously reads multiple ascii files (.dat or .txt), each with three data columns (tab- or space-separated) representing the three partial fluorescence traces ( $PF$ , i.e. the FRET efficiency in a multi-color experiment) of one molecule. File names are arbitrary, as the imported trajectories are renamed during import. The original names are stored in Igor’s “WaveNotes”.

Input data may include NaNs or INFs. It should only contain the range that is relevant for analysis (e.g. no after-bleach tail). The supplied input data determines the time units of all calculations: if your sampling rate is  $x$  frames per second, all transition probabilities are specified per time interval  $\Delta t = 1/x$ . Therefore, constant time intervals are required. But specific time information is not needed.

## Step 1: Population Selection and 3D Histogram Fitting

Remove molecules that show a low signal-to-noise ratio in the  $PF$  traces. Molecules that exceed the interval  $[-1;2]$  in any  $PF$  trace for more than 10% of the frames are removed from the data set. Execute

```
>> RemoveTracesLowSNR()
```

Calculate binned 2D projections of the  $PF$  data. Execute

```
>> HistFret2D("r_b", "r_g", binHist=100)
```

and

```
>> HistFret2D("r_b", "g_b", binHist=100); MoveWindow 553.5, 42.5, 1055.25, 508.25
```

These commands plot  $PF_{red}^{green}$  over  $PF_{red}^{blue}$  and  $PF_{green}^{blue}$  over  $PF_{red}^{blue}$  in the range  $[-0.5; 1.5]$  with a resolution of 100 x 100 bins.

Now, estimates for the population of each distinguishable state have to be found. Bring the appropriate 2D histogram to the front and execute

```
>> panelHist2DCount()
```

Press the “Init” button and draw a free-hand polygon around the peak. Click the “Count” button. The number of data points in the polygon and the total number of data points in the projection are printed in the Command Window.

Prepare a 3D histogram of the  $PF$  data. Execute

```
>> HistFret3D("g_b", "r_b", "r_g")
```

Normalize the 3D histogram to an integral of 1. Execute

```
>> NewDataFolder/S fit0
```

```
>> Duplicate/O ::FRET:Hist3D, Hist3D
```

```
>> Variable /G div = sum(Hist3D)*(DimDelta(Hist3D,0))^3
```

```
>> Hist3D /= div; Print div
```

Provide initial parameters for the 3D Gaussian fit and prepare necessary data structures. Execute

```
>> Gauss3D_initParam(); edit W_coef_old
```

and add the state populations to the end of the parameter vector. For the provided test data set, the values should be close to {0.57, 0.13, 0.17, 0.11, 0.02} for states s0 to s4.

Note: W\_coef\_old is a vector that holds the initial parameters for the fit. Per state this means  $\mu_x, \mu_y, \mu_z, \sigma_x^2, \sigma_{xy}, \sigma_{xz}, \sigma_{yx}, \sigma_y^2, \sigma_{yz}, \sigma_{zx}, \sigma_{zy}, \sigma_z^2$  and the state population, which is concatenated at the end of the vector. Please keep in mind that the co-variance matrix has to be symmetric. The parameters preset in the software are a good starting point for the supplied example data set and have to be adjusted accordingly if you are working with your own data.

Prepare all data structures for the fit by executing

```
>> Gauss3D_prepareFit()
```

Fit the sum of  $S$  3D Gaussian functions to the 3D *PF* histogram, with  $S$  being the number of distinguishable states (in this case  $S=5$ ). Execute

```
>> do3D()
```

```
>> postprocessFitMultiGauss3D(); evalFitMultiGauss3D(); edit W_coef
```

Display the fit result. For each of the two 2D projections, use the commands

```
>> contourPF3D_new(0);contourPF3D_new(1);contourPF3D_new(2);contourPF3D_new(3);
```

```
contourPF3D_new(4)
```

```
>> contourPF3D_colorize()
```

## Step 2: Kinetic Analysis with 3D Ensemble HMM

### Converging of the 3D HMM

Initialize the HMM user interface (“3D HMM” | “Init HMM”) and choose the appropriate number of states (in the case of the Hsp90 data this means “NumStates” = 5), number of dimensions of the input signal (“NumDims” = 3) and the type of the input (“Input Type” = “FRET 3D bgr”).

Converge the HMM with

```
>> prepENS_CONVERGE_gB(GetDataFolder(1), -14)
```

Note: Optimization stops when the change in the transition matrix compared to the previous iteration falls below a threshold ( $10^{-14}$  for the sum of the absolute change for each transition probability).

### Confidence Intervals

The confidence intervals for the transition probabilities report on the data set heterogeneity and the precision of the HMM. A rough estimate of the bounds of the confidence interval is found by executing

```
>> cd $(root:path3Dimport + "HMM"); loop_getCI_estimate_limits()
```

The 95% confidence bound for each rate is reached when the likelihood ratio exceeds 3.841, the 95% quantile of a  $\chi^2$ -distribution with one degree of freedom. Execute:

```
>> loop_getCI_HMM_converge(1)
>> CIresults_conv_new()
>> cd ::HMM_CIresult; reportCI_conv()
>> cd ::cmp_CI_conv; CI_plot2("HMM", doAppend=0)
```

## Interpretation of the Kinetic Analysis

In order to collapse states (e.g. because they are functionally identical) use

```
>> cd ::HMM
>> collapse_states_get_DT({0,1,1,1,0})
>> plot_collapsed_DT_Hist (wDTo_01110_record1)
```

Note: These commands accumulate dwells of states s0 and s4 as well as dwells of states s1, s2 and s3 and plots a dwell time histogram of the combined dwells of s1, s2 and s3.

## Data Export

The resulting Viterbi paths for each molecule, a summary of the HMM run and the confidence intervals can be exported by using

```
>> Export_Results()
```