# Analysis of protein stretching experiments

The GitHub repository at

<https://github.com/are-mj/ProteinStretching>

contains a set of Matlab functions and other tools for analysing protein stretching experiments using optical tweezers. Below is a summary of the main functions and how to combine them in a workflow starting with text files with raw experiment text files and ending with estimates for the model parameters , and .

## 1 Personalise environment.

The following files must be modified to match your folder structure:

**datafolder.m**  
Modify the path to the folder where your experiment \*.txt files are stored.

**myfiles.m**  
Modify the list of files you want to analyse together (e.g., with analyse\_many)  
You may have several such list for different purposes. Use informative names for these files, e.g., Top7files.

**temperature\_code.m**

Make sure that your experiment files are assigned to the appropriate list. Add new lists id needed.

## 2 Finding unfolding/refolding events

### analyse\_file.m

Basic inputs:

* Experiment record file, e.g., <datafolder>\07182022\bA.txt  
  where <datafolder> is the output of datafolder.m.
* COM file, e.g. <datafolder>\07182022\bCOM.txt

Remember to modify datafolder.m to match your installation.

Analyse file to find unfolding and refolding events:

[Tu,Tr] =analyse\_file(file[,plotting])

* Inputs:
  + file (e.g ‘07182022\bA.txt’ or the full path
  + plotting: Optional. 1 to plot f(t) with identified events. Default: 0
* Ouputs:
  + Matlab tables Tu (unfolding events) and Tr (refolding events)

Plotting is recommended. This lets you inspect the quality of the identification of events.

**Tables Tu and Tr** from analyse file.

The outputs Tu and Tr from analyse\_file are Matlab tables with one row for every identified unfolding or refolding event, respectively. The tables have the following columns:

* Filename
* Time event time(s).
* Deltax x distance between f curves before and after event (nm),
* Force Force at transition event (pN).
* Forceshift Force change at transition event (pN).
* Fdot Force rate of change before event (pN/s)-
* Pullingspeed Speed of trap position during main part of stretching or relaxing (nm/s)
* Temperature Temperature at transition event (°C).
* dFdx Mean slope of force vs extent before unfolding or refolding event.
* dt: Sampling time (seconds/sample)
* Lineno Line number in experiment file when transition occurs.

Analyse many files at once.

[Tun,Tre] = **analyse\_many**(Filenames);

Filenames may be an array of Matlab strings or a function that returns such an array. Examples may be myfiles or Top7files.

You can write a table to Excel using **writetable**:

writetable(Tun,’Top7 experiments.xlsx','Sheet',’All')

## 3 Probability density distribution

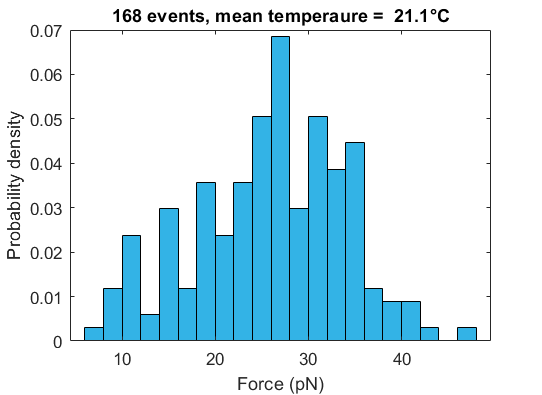
### probability\_density.m

Calculates histogram values of probability density pd vs. unfolding force F. The input is a table produced by analyse\_many. It is good practice to extract groups of experiments where pulling speed and/or temperatures are not too dissimilar. Use **select** to extract relevant rows in the table.

The unit for probability density is pN-1.

T0 = select(Tun,'20<Pullingspeed<200 & 19<Temperature<22');

[pd,F,n,Fdotmean] = probability\_density(T0,2,1);



The output from probability\_density may be used as input to the model parameter fitting functions.

## 4 Estimating model parameters

### fit\_unfold\_parameters

This function uses nonlinear least squares optimisation to fit the output of **unfold\_probability\_model** to the experiment values. See below fir details.about the parameters

[theta,theta\_std,resnorm,Fplot,pdplot] = ...

fit\_unfold\_parameters(F,pd,T,theta0,par)

See the help text for fit\_unfold\_parameters for details about inputs and outputs.

**Parameter confidence intervals.**

The 95% confidence interval for the parameters are calculated by nlparci (Non-Linear Parameter Confidence Interval) from Matlab’s Statistics and Machine-Learning Toolbox. We translate this to the perhaps more familiar standard deviations.

**unfold\_probability\_model**

This is an implementation of The Dudko model for unfolding probability density as a function of force (Dudko, Hummer and Szabo, 2006) . The model uses the following expression for the reaction rate:

Based on this rate the Dudko model calculates unfolding probability density as a function of force . We wish to find estimates of the parameters , and so that the model matches the experimental probability densities as well as possible.

In the Matlab code we use DG and dx for and , respectively. We also use log10(k0) instead of , so the parameter vector is:

theta = [DG; dx; log10k0]

The probability densities should not take complex values, so to keep the expression inside the parenthesis positive. Fit\_unfold\_parameters therefore use a transformed parameter vector:

theta\_calc = [DG; a; log10k0]

as input to this function. The modified parameter vector lets us specify an upper bound on a: , ensuring real-valued variables. is the maximum F value from probability\_density.

### fit\_refold\_parameters

This function is similar to fit\_unfold\_parameters but uses the Bell model, which is equivalent to the Dudko model with . This is independent of and thus has only two parameters:

theta = [x; log10k0];

The model is implemented in **refold\_probability\_model**.

## Example

[Tun,Tre] = analyse\_many(myfiles,1);

Tun\_fast = select(Tun,'Pullingspeed>500');

[pd,F,n,Fdotmean] = probability\_density(Tun\_fast,1,1);

par.model = 'DHS';

par.nu = 0.5;

par.Fdotmean = Fdotmean;

T = 25;

theta0 = [50;10;-5];

[theta,theta\_std,resnorm,Fplot,pdplot] = ...

fit\_unfold\_parameters(F,pd,T,theta0,par);

To show the fit and the parameters:

hold on; plot(Fplot,pdplot,'r','linewidth',2);

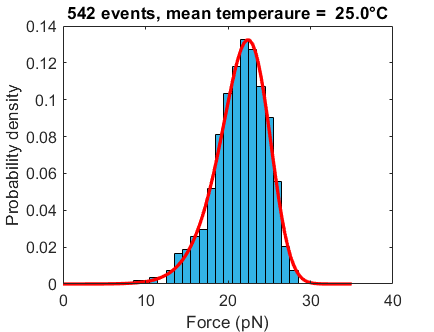
fprintf('%10s %10s %10s %10s\n',' ','DG','dx','log10(k0)')

fprintf('%10s %10.2f %10.2f %10.2f\n','Parameters:', ...

theta(1),theta(2),theta(3))

fprintf('%10s %10.2f %10.2f %10.2f\n','STD:', ...

theta\_std(1),theta\_std(2),theta\_std(3))



DG dx log10(k0)

Parameters: 67.76 3.26 -3.93

STD: 2.05 0.46 0.50

# File list

|  |  |
| --- | --- |
| **File name** | **Purpose** |
| analyse\_file.m | Identify unfolding/refolding events |
| analyse\_file\_demo.mlx | Interactive demo (needs revisions) |
| analyse\_many.m | Runs analyse\_file for a list of files |
| analyse\_trace.m | Analyse a single trace. Used by analyse\_file. |
| analyse\_trace\_demo.mlx | Interactive demo (needs revision) |
| convert\_energy.m | Conversion from zJ/molecule to kcal/mol |
| datafolder.m | Path to the experiment data files |
| display\_trace.m | Display trace corresponding to a point |
| dominant\_frequency.m | Helper function to identify force peaks and valleys |
| extract\_trace.m | Extract trace from timestamp |
| fit\_refold\_parameters.m | Fit model parameters to refold data |
| fit\_unfold\_parameters.m | Fit model parameters to unfold data |
| movingslope.m | Helper function for analyse\_trace |
| myfiles.m | Return list of data files |
| probability\_density.m | Probability density histogram from tables |
| read\_experiment\_file.m | Used by analyse\_file |
| refold\_probability\_model.m | Bell refolding probability model |
| select.m | Select table rows satisfying given criteria |
| stretch\_relax.m | Extract pair of corresponding stretch and relax traces |
| temperature\_code.m | Coding tables for reading temperature from status column |
| traces.mat | Set of traces for testing or demonstration |
| unfold\_probability\_model.m | Dudko's model for unfolding probability density |
| valid\_data\_ranges.m | Specification of parts of files that can be analysed together |