# Tools for Analysis of protein stretching experiments.

2023-10-24

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**

It is common to store temperature as a code in digits 2 and 3 of the Status column. Make sure that your experiment files are assigned to the appropriate list. Add new lists as needed.

If temperature is not recorded in the Status column, analyse\_file will look for the appropriate COM file and read temperature there.

## 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("02032022/cA.txt",1);

xlim([0,350]); ylim([0,35])

A graph of a graph

Description automatically generated

The second input argument is optional. Set it to 1 to plot the result. Plotting lets you inspect the quality of the identification of events. Red dots are unfoldings, black circles are refoldings.

**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 (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. Example:

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

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

writetable(Tun,'My 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.

The unit for probability density is pN-1.

It is good practice to analyse table rows with similar temperatures and Pulling speeds together. This can be done by the **select** function:

T0 = select(Tun,'Pullingspeed>500 & 20<Temperature<30');

dF = 1;

[pd,F,n,Tmean,Fdot] = probability\_density(T0,dF);

figure;

bar(F,pd,1)

xlabel('Force (pN)')

ylabel('Probability density (pN^-^1)')

A graph of a number of blue bars

Description automatically generated

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

## 4 Estimating model parameters

### fit\_Dudko\_unfold, fit\_Bell\_unfold

These functions use nonlinear least squares optimisation to find parameters that fit the output of **Dudko\_unfold\_probability** or **Bell\_unfold\_probability** to the experiment values.

Initial values of the Dudko parameters may be:

is the rate of unfolding at zero force, which is very low, so a good initial value may be:

par.nu = 0.5; par.model = 'DHS';

theta0 = [50;5;-4]; % Initial parameter vector

[theta,theta\_std,resnorm] = fit\_Dudko\_unfold(F,pd,Tmean,Fdot,theta0,par);

Dudko\_unfold\_probability uses a transformed version (thetacalc) of the parameter vector (theta) as input:

thetacalc = [theta(1);par.nu\*theta(2)/theta(1);theta(3)];

Fplot = linspace(5,30); hold on;

plot(Fplot,Dudko\_unfold\_probability(thetacalc,Fplot,Tmean, ...

Fdot,par),'r','linewidth',2)

A graph with a red line

Description automatically generated

**Why do we use the transformed parameter vector?**

The Dudko unfolding rate is:

The normal parameter vector is

By replacing by , we can set an upper limit which ensures that the expression is never negative during the search for the optimum, thus avoiding unphysical complex values that will crash the calculation.

We use as parameter rather than , partly out of convenience, but also because it makes more sense to express standard deviations in terms of the logarithm.

You may display the results by, e.g.:

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))

This gives:

DG dx log10(k0)

Parameters: 67.95 3.29 -3.96

STD: 2.13 0.46 0.50

The **Bell** model uses only the two parameters and , so it cannot be used to determine . But **fit\_Bell\_unfold** is more robust than **fit\_Dudko\_unfold**.

**Parameter confidence intervals.**

The parameter fitting functions calculate the 95% confidence intervals for the parameters using nlparci (Non-Linear Parameter Confidence Interval) from Matlab’s Statistics and Machine-Learning Toolbox. We translate this to the perhaps more familiar standard deviations.

### fit\_Bell\_refold

At present, this is the only model to describe refolding. The analysis goes analogous to the unfold case, with one or two modifications:

Since the span of refolding forces is smaller than for unfolding, a force spacing of 0.5 may give nicer bar graph.

Since is the rate of refolding at zero force, it has a large value. A sensible initial parameter may be

theta0 = [3;5];

Fplot = linspace(5,15);

T0 = select(Tre,'Pullingspeed>500 & 20<Temperature<30');

dF = 0.5;

[pd,F,n,Tmean,Fdot] = probability\_density(T0,dF);

bar(F,pd,1)

[theta,theta\_std,resnorm] = fit\_Bell\_refold(F,pd,Tmean,Fdot,theta0);

hold on;

plot(Fplot,Bell\_refold\_probability(theta,Fplot,Tmean, ...

Fdot),'r','linewidth',2);

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

fprintf('%10s %10.2f %10.2f\n','Parameters:',theta(1),theta(2))

fprintf('%10s %10.2f %10.2f\n','STD:',theta\_std(1),theta\_std(2))

A diagram of a normal distribution

Description automatically generated

dx log10(k0)

Parameters: 6.45 8.45

STD: 0.11 0.12

# File list

|  |  |
| --- | --- |
| Name | Purpose |
| analyse\_file.m | Finds unfolding and refolding events in experiment file |
| analyse\_many.m | Runs analyse\_file for a set of files |
| analyse\_trace.m | Finds unfolding and refolding in single trace |
| Bell\_refold\_probability.m | Unfold force histogram from table |
| Bell\_unfold\_probability.m | Refold force histogram from table |
| convert\_energy.m | Convert from zJ/molecule to kcal/mol |
| datafolder.m | Path to folder for experiment results file |
| display\_trace.m | Display traces marked by data tips in scatter plot |
| dominant\_frequency.m | Find the dominant frequency in a time series |
| Dudko\_unfold\_probability.m | Model probability density for unfolding. |
| extract\_trace.m | Extract one trace, given filr name and time |
| fit\_Bell\_refold.m | Fit parameters for Bell refolding model |
| fit\_Bell\_unfold.m | Fit parameters for Bell unfolding model |
| fit\_dual.m | Fit parameters for dual unfolding or refolding model |
| fit\_Dudko\_unfold.m | Fit parameters for Dudko unfolding model |
| movingslope.m | John D'Errico's function for calcuating slope in a moving window |
| myfiles.m | Sample file list |
| probability\_density.m | Probability density histogram from table |
| probdens.m | Probability density histogram from force array |
| read\_experiment\_file.m | Read relevant coluns from experiment file |
| README.md | Description of repository |
| run\_dual\_model.m | Running fit\_dual for a set of temperatures and pullimg speeds |
| select.m | Select rows in Matlab table that satisfy one or more criteria |
| T\_from\_COM.m | Read temperature from COM file (if relevant Tlist not defined |
| temperature\_code.m | Returns relavant temperature coding table |
| Top7files.m | Returns list of good Top7 experiment files |
| traces.mat | Set of trace data for illustration |
| User guide.docx | This document |
| valid\_data\_ranges.m | Helper function for analyse\_file |
| wlc.m | Worm-Like\_Chain model - force(extent) |
| wlc\_inverse.m | Worm-Like\_Chain model - extent(force) |