# Experimental data and program files

for the paper

**Top 7 is a metamorphic and a possible prion-like protein**

## Overview

Programs for analysing the data files from the experiments are written in Matlab. Typically a batch of experiment files are analysed by function analyse\_many, which returns Matlab tables Tun (unfoldings) and Tre (refoldings). These tables contain twelve recorded values per trace, most notably the time (seconds from experiment start), rip or zip force (pN), the rip/zip length Δx (nm), the temperature (°C) and the pulling speed (nm/s).

The tables for all experiments are stored in the file Top7tables.mat and are used to generate most figures and tables in the paper and supplement.

Each of the tables and figures are generated by a separate m-file named after the figure or table.

## Data files

Time series of experimental recordings are stored in folders named after the experiment date. There may be several text files in each folder. File names typically have names on the form aA.txt, where the lowercase letter(s) signify the individual molecule. For long experiments, results may span several files, identified by sequential capital letters. The files may have varying number of columns. The columns used are :

CycleCount: Time measure relative to experiment start. 400 cycle counts per second

Y\_force: Pulling force (pN)

A\_dist-Y, B\_dist-Y: Position of pulling bead (nm) (trap position). We use the mean of the two.

Status: Used to code the temperature in the chamber. See temperature\_code.m.

For completeness we also include data that for various reasons were excluded from the analyses. The reasons for exclusion were not systematically recorded but they include incorrect experiment setup, disturbances and too much noise. These files are found in subfolder ‘excluded\_data’.

## Automatic analysis of experimental raw data

The experiment text files are read by read.experiment\_file.m which returns t (time, seconds), f (force, pN). x (trap position, nm) and temperature T (°C).

Analyse\_file.m uses this information to find individual stretching and relaxing traces. Each trace is analysed by analyse\_trace.m which looks for changes in the slope of the force trace to identify candidates for unfolding (rip) or refolding (zip) events. The candidates must fulfil a set of requirements to qualify and the best of the qualified candidates is chosen. analyse\_trace returns a Matlab struct with the rip/zip properties. analyse\_file uses this to create one row per event in the output tables Tu and Tr. A number of parameters have been tweaked in order to maximise detection efficiency, with a higher priority given to avoiding spurious events. The same parameters were used for all experiments.

Specify plotting to see a graph of the force time series with rips and zips marked. Example:

analyse\_file("20230623/fA.txt",1);

This file is from an experiment with three different pulling speeds. Use the zoom and pan tools that appear at the upper right to see more detail.

#### Post-processing

analyse\_many.m calls analyse\_file repeatedly for all file names returned from Top7files.m, Top7Top7files.m or Top7BSAfiles.m. The two latter contains filenames for the experiments with proteins in solution. Some results for Top7files had to be modified to compensate for calibration errors in a small subset of experiments. Thus is done by script correct\_bias.m. Furthermore, the output table from Top7Top7files contains one outlier unfolding at > 70pN. This was removed before storing to Top7tables.mat.

## Clustering

The categorisation of events as belonging to Cluster 1, Cluster2 or as outliers was done by function no\_outliers.m, which uses the Matlab clustering algorithm dbscan to assign points in a scatter plot to groups based on distances between neighbouring points. The parameters were chosen by trial and error until the results seemed reasonable.

## Grouping experiments and fitting models to data

The experiments were grouped into 14 unfolding and 8 refolding groups based on temperature and pulling speed. The Bell and Dudko models calculate force probability density functions. The Crooks model uses histograms of unfolding and refolding work instead. For each group and cluster the models were fitted to the data using Matlab’s lsqcurvefit function. The grouping and fitting were done in the function run\_fit\_dual.m. The resulting parameters were reported in the output table Tout. This table was used to generate most tables and graphs.

## Matlab details

The following Matlab toolboxes are required:

|  |  |
| --- | --- |
| Toolbox | Used by |
| Optimization | Fit\_Bell\_unfold  Fit\_Bell\_refold  Fit\_Dudko\_unfold |
| Signal Processing | Anayse\_file  Analyse\_trace  Dominant frequency |
| Statistics and Machine Learning | Fit\_Crooks  Fig\_S7  No\_outliers  Fit\_Bell\_unfold  Fit\_Bell\_refold  Fit\_dudko\_unfold |

All files were tested in Matlab R2024a. Most were also run in Matlab R2023a.

## Acknowledgement

movingslope.m was created by John D’Errico and was copied from the Matlab Central File Exchange.

Ref.: John D'Errico (2024). Movingslope (https://www.mathworks.com/matlabcentral/fileexchange/16997-movingslope), MATLAB Central File Exchange. Retrieved April 19, 2021.