Auto force curve analysis

Xin He, Rice University

**[Abstract]** At present, data analysis of AFM research mainly relies on manually operating various software. Taking the Tether Force analysis as an example, it usually takes about one hour to analyze the results of a single experiment. At the same time, because of the differences in individual standards, different people may obtain different results when analyzing the same data. In this paper, we have developed a software that can automate the analysis of AFM Force curve data through mathematical and signal processing. This software implements one-click data processing and gives more analysis results of the Force curve. It has been used by our experimental group for routine data analysis and has been shown to significantly increase the efficiency of analysis and eliminate the differences in human manually processing.

**[AFM Output data and preprocessing]**

In the AFM cell tether force experiment, we get the curve of the sensing signal with respect to time. The pre-processing transformed the original data into the force-distance curve, which is the so-called force curve we are going to process. First of all, our software uses this curve as the initial two input vectors.

**[Algorithm]**

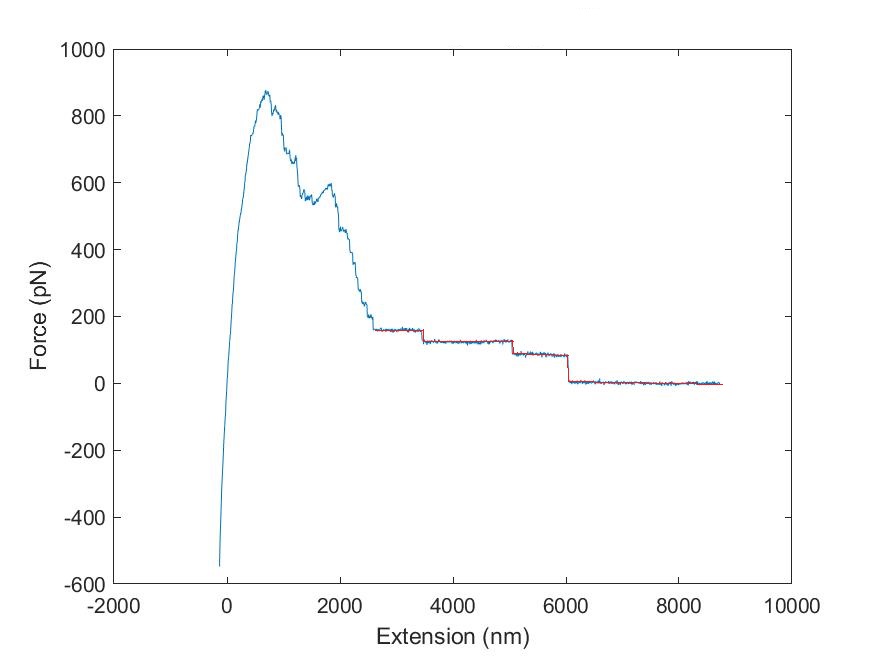
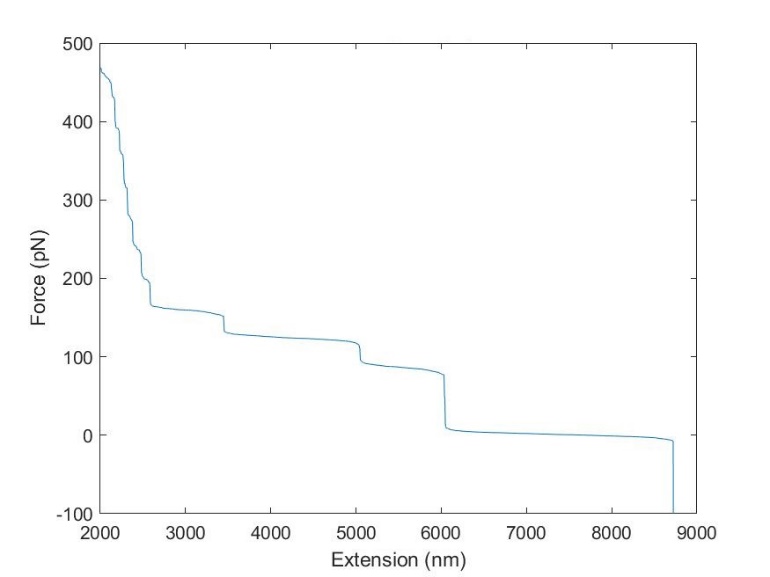
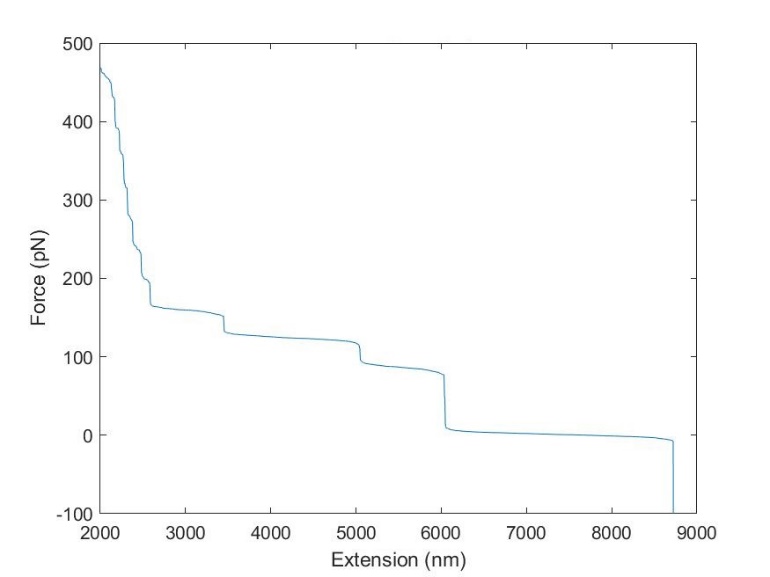


Figure 1. One of the force curve of cell line MCF-10A. The selected red part is the aim force step.

First of all, we need to standardize each curve, that is, looking for the baseline. Here we start scanning from the end of the curve until we find a straight segment. And then use this segment as a benchmark to standardize the entire curve. If we are not able to find a baseline that meets the requirements. That means this data is unstable and we prefer to skipping it.

After normalization, the program intercept the latter falling part where might be force steps of the curve. Since the step force we are looking for only concerns about the data of Y-axis, we may wish to sort the intercepted curve with Y-axis data as the key word. This will upset the original order of the X-axis, but will not affect the final result of step force. At the same time, by sorting, we can get a smooth monotonic curve with the original gaussian noise reflected in the change of slope (figure 2a).

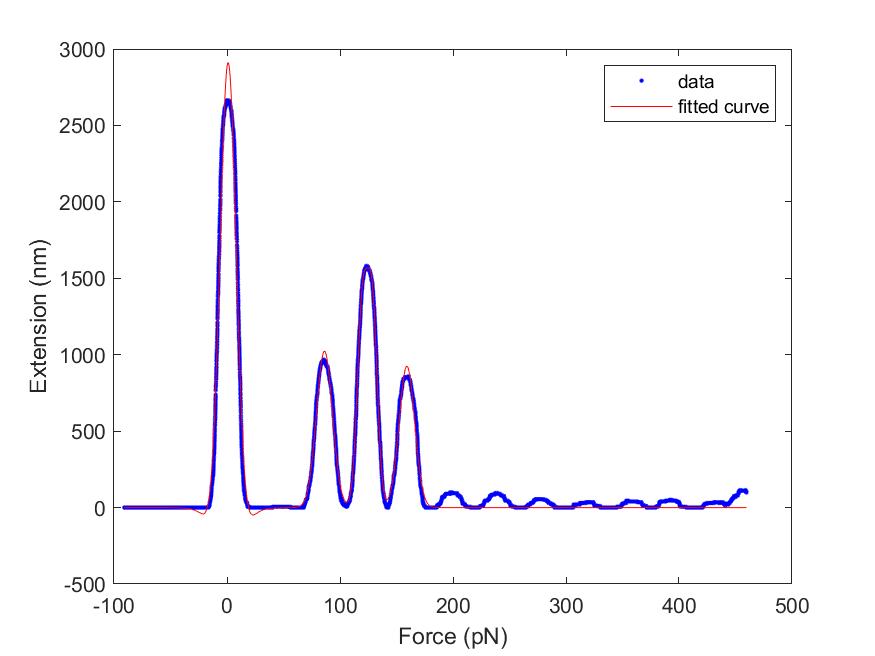
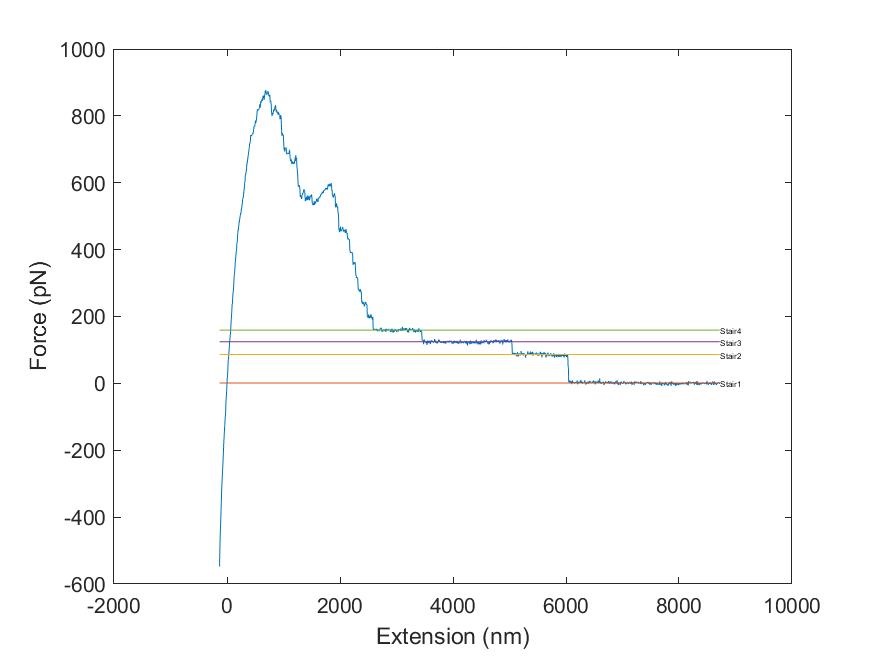


**(b)**

**(a)**

Figure 2. (a) Sorted curve of figure 1. (b) To obtain the location of the step, use a ruler-like rectangular to scan the curve and record the horizontal length.

Then, we use a long rectangular box to do a step-by-step scan from the bottom of the curve (figure 2b), recording the X-axis length of the curve contained in the frame each time. After this step, we can get a curve about the length with respect to force. The principle of obtaining this curve is similar to the distribution of the original image using Y-axis data as the key. In the graph obtained in this step, we can clearly see that the step we were looking for has become a single Gaussian distribution peak. The reasons for this phenomenon are as follows. The step in the original figure is that the force remains constant for a certain pulling distance. Due to the accuracy of the experimental equipment, the flat straight segment we get will have gaussian noise. If we generate the data point distribution for this step, we will get a gaussian distribution, and the middle value of this distribution is the force. The rest parts which are not steps are almost not displayed or displayed as other irregular curves on the resulting map due to the distribution. Therefore, as long as we transform the force curve with the above steps, we can locate the positions of the steps by gaussian fitting and the X-axis distance between the gaussian peaks is the step force we want to calculate.

According to the discussion above, the next process is obvious and logical. We performed a multi-peak Gaussian fit on the resulting graph to obtain the position of each peak, and then we performed a more accurate single-peak Gaussian fit on each individual peak and calculate their R-square values. When the R-square value meets the set requirements which means the fitting is good enough, the program takes this peak as a real step. At the same time, the height of the peak can tell us the information about the horizontal length of the step. Here we set a height minimum to prevent the program from counting some short segments.

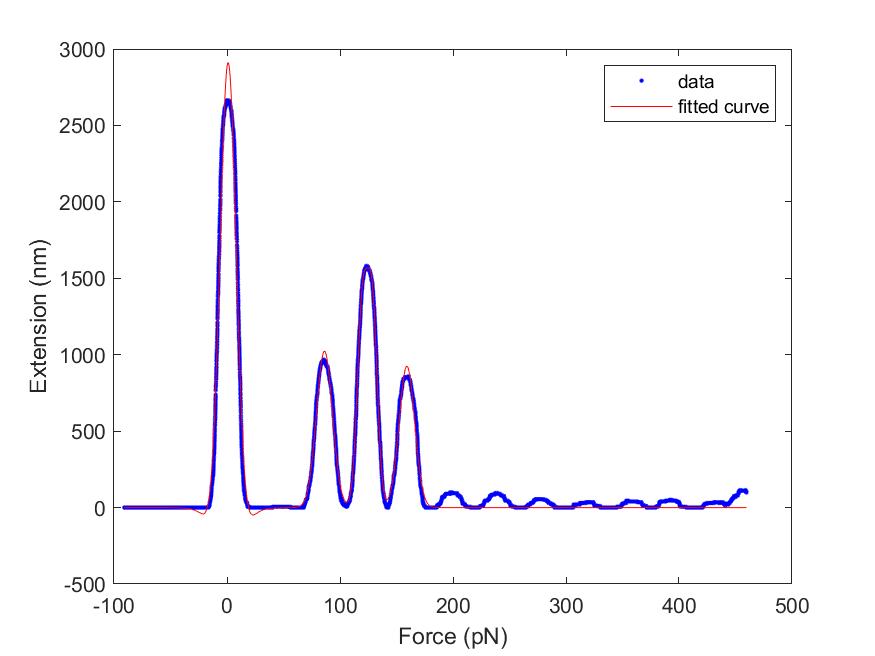


Figure 4. The analyzed curve presented by the program in manually review part. Straight line with different color used to mark the stairs.

**(b)**

**(a)**

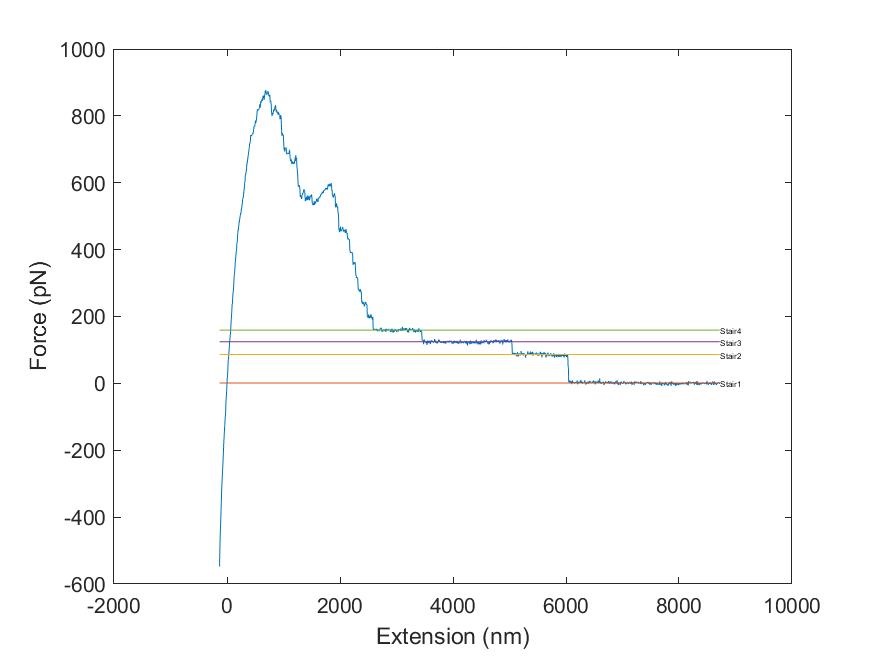
Figure 3. (a)Scanning result of figure 2a, the red marked curve is the result of multiple-peaks gaussian fitting. (b) The X-axis distance between the gaussian peaks is the step force we want to calculate.

After locating the position of the step, the step force is calculated as long as the difference between the adjacent steps is calculated. The program repeats the above steps for all data, and finally gives a histogram of the step force.

All automated analysis is not guaranteed to be 100% accurate, let alone an analysis of the step force with obscured standard. The error of the above algorithm is mainly come from the error of multi-peak gaussian fitting. To reduce errors, we added an optional manual review part at the end of the program to enhance the accuracy of the program. In this part, the program will display the data it has selected for the step and will mark the location of each step. When there is an error, the researcher can manually remove the wrongly selected step.

**[Tether Lifetime]**

Figure 5. Lifetime of each tether can be measured from the force curve



In the above calculation process, we can actually get a parameter which is the length of the step, as shown on figure 5. When we convert back to the original data, we can get the lifetime of each tether. This program can generate the overall histogram. What we get after normalization is actually a probability density image. A probability distribution curve can be obtained after integrating the obtained probability density. Here we use the exponential decay model to fit it to get the average lifetime of the tether.