Temperature dependent protein unfolding

Trevor King

June 1, 2007

Abstract

Protein structure prediction is a central problem in current biophysics. Understanding the energy landscape roughness would shed light on the physics behind protein folding. The theory for extracting energy landscape roughness from temperature dependent unfolding measurements is reviewed. Improvements facilitating temperature dependent single molecule protein unfolding experiments are described. We discuss the automation of AFM stepper motor control, and present an experiment server and an automatic force curve sorting algorithm.

1 Introduction

Proteins are polymers of amino acids of critical importance to biochemistry. They play a role in almost every aspect of life, from catalyzing reactions in a cell to forming the mechanical structure of fingernails, muscle, and hair. A thorough understanding of the formation and action of proteins is fundamental to a theoretical understanding of life's inner workings.

The amino acid sequence for a given protein is encoded in DNA. With the completion of the human genome project and the approach of fast DNA sequencing, there is an abundance of information about these sequences. However, moving from a protein's amino acid sequence to its biological function or even its three dimensional structure is currently very difficult.

Protein structure can be measured experimentally using x-ray crystallography or NMR spectroscopy, and such work is currently filling databases of protein structures. This horizontal cataloging should be supported by a vertical attempt to study a smaller group of proteins in detail and understand not only their conformation, but *why* they assume that conformation. A firmer grasp of the physics behind the protein folding process would almost certainly aide in the push towards predictive folding models.

1.1 The protein folding problem

Because protein folding involves finding a global energy minimum in a convoluted high dimensional-space, it is very difficult to model. As Levinthal pointed out in 1969, the number of possible conformations a protein can assume is so large that searching all of them to find a lowest energy state would take many times the age of the universe[1]. The local roughness in the energy landscape of a protein which shields the protein from the global minimum energy conformation is referred to as *frustration*.

Because proteins in nature *do* fold in reasonable timescales, their energy landscape cannot be a uniform bumpy surface, but must be roughly funnel shaped, guiding the proteins toward the more stable conformations (see Figure 1). This makes predicting the folded state of native proteins easier, and sets design goals for stable artificial proteins.

In this project, we attempt to probe this energy landscape for the muscle protein Titin. With more information about the general structure of the folding funnel, we can perhaps construct more appropriate theoretical tools for seeking stable conformations.

1.2 Single molecular folding and unfolding

Most methods for studying protein folding and unfolding use bulk samples containing many molecules.. The proteins are unfolded by changing the solution pH, temperature, and other environmental factors. The state of the protein is monitored using florescence, NMR, or other methods[2]. Such

ensemble measurements allow parallel measurements of millions of unfolding events, which leads to very convincing average values of the measured parameters but loses any information about their distributions.

Single molecule folding and unfolding experiments are slower and more difficult, but they give access to distribution of protein behaviors, which may yield information not present in the averages. The direct mechanical manipulation methods used on single molecules also benefit from not effecting the protein environment, so temperature and pH can be chosen to match those expected *in vivo*.

1.3 Temperature dependent unfolding and the energy landscape

One limitation of the single molecule protein unfolding experiment is that we only directly measure how the unfolding force depends on the single parameter of end-to-end distance. Varying some environmental factors, and measuring how the unfolding forces depend on environmental conditions can provide deeper insight into the mechanics. For example, measuring how the unfolding force for a given protein varies with respect to temperature and pulling speed can yield a scale for the energy landscape frustration (see Figure 1), shown by Hyeon and Thirumalai[3]. This theory forms the basis for the the bulk of our experiment, and is discussed in detail in Section 2.7.

1.4 Titin and temperature dependent muscle behavior

Measuring how the unfolding force of Titin depends on temperature not only probes its energy landscape, but might also shed light on why muscle stiffness changes with temperature. Titin is one of three main proteins in striated muscle along with Actin and Myosin. While Actin and Myosin are responsible for contractions, Titin is responsible for the return to the relaxed state. Titin is also responsible for the mechanical properties of the relaxed state (see Figure 2)[4]. Because of Titin's biological role, our forced mechanical unfolding procedure closely mimics the actual experience of Titin *in vivo*. We hope an increased understanding of the fundamental temperature dependent folding properties of Titin will shed light on the temperature dependent mechanical properties of muscle.

2 Experimental setup and procedure

We mechanically unfold our proteins by pulling them apart with an atomic force microscope (AFM). A brief review of the operating principle of an AFM is given in Section 2.1. We discuss the procedure used to unfold a protein, give an overview of the supporting equipment, and present some example data in Section 2.2. A temperature control module was built by a former graduate student in order to monitor and control the temperature at which we unfold our proteins. We will discuss the temperature control setup in detail in 2.3.

(a) A random landscape[5].

(b) A funnel landscape[5].

Figure 1: (a) The energy landscapes of most random polypeptide chains are rough with no unique global minimum. (b) Native proteins have a small number of stable minima and an overall funnel shape, which encourages rapid folding into biologically useful conformations[5].

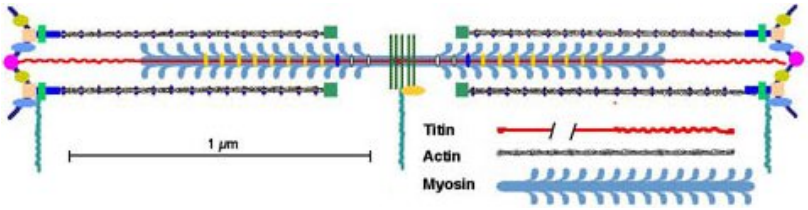


Figure 2: The role of Titin in a striated muscle sarcomere[6]. Titin keeps the Myosin centered between Actin filaments and is the source of passive stiffness when the Myosin is not bound to the Actin.

We have been streamlining our experiment to ease the collection of the large datasets required to extract statistically significant amount of data. In preparation for temperature-dependent protein unfolding experiments, we started making some alterations to our experiment control software and hardware. One goal was to obtain automatic control of the stepping motor used for coarse positioning, which will allow us to compensate for thermal drift thus extend experiment run times. We discuss the automatic stepper-control system in Section 2.4.

As the stepping motor software evolved, we had difficulty updating the experiment control software to integrate the new features. After much thought, we decided that the best long term solution would be to scrap our current architecture and rebuild our software in a more modular and flexible fashion. We discuss our resulting experiment server project in Section 2.5.

In order to extract useful information from the a protein unfolding experiment, the successful protein unfolding curves must be separated from an abundance of force curves without unfolding events. We discuss a developing procedure for automatically sorting force curves in Section 2.6.

Finally, we discuss the theory that allows the extraction of frustration from temperature dependent unfolding measurements in Section 2.7, and apply it to some previous Ubiquitin unfolding data in Section 2.8.

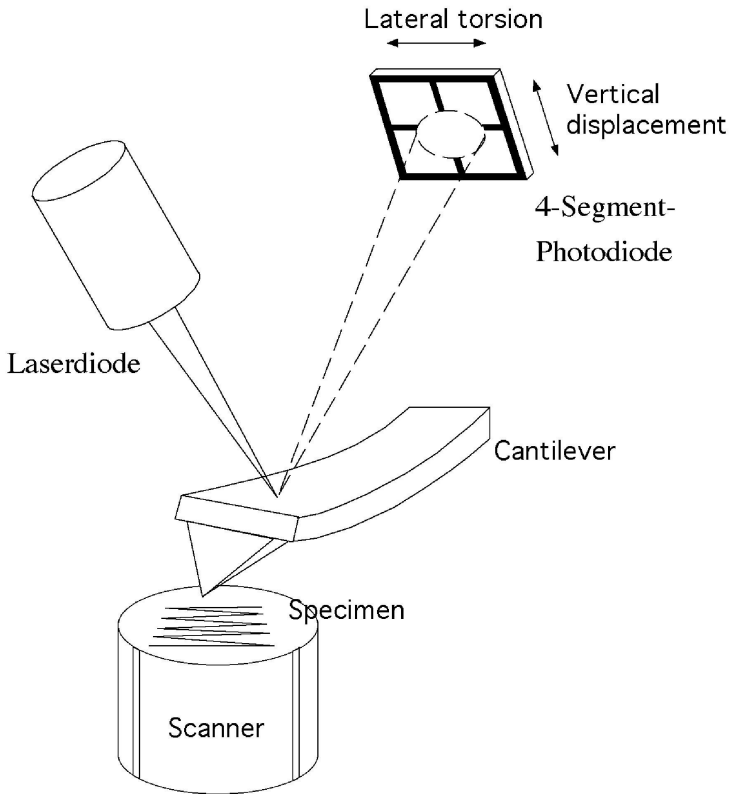


Figure 3: Basic AFM operation[7]. The laser bounces off of the reflective back of the cantilever and hits a four-segment photodiode, which records the location of the laser spot. The force on the tip can be measured indirectly by watching the deflection of the laser spot and modeling the cantilever as a spring obeying Hooke's Law. The specimen is mounted on a piezoelectric scanner, which allows it to be positioned in three dimensions relative to the tip. In our setup, the entire cantilever-surface system is enclosed by a buffer-filled fluid cell.

2.1 Atomic force microscopy

We obtain unfolding forces for proteins by mechanically pulling with an AFM (see Figure 3). The AFM gives us access to picoNewton forces over nanometer distances, which allows the unfolding of individual protein to be monitored directly.

Because AFMs have traditionally been used for imaging, commercial control systems and software are geared towards those applications. Our AFM control hardware (NanoScope IIIa, from Digital Instruments, Santa Barbara, CA) has been significantly modified to give direct access to the piezo control lines and photodiode signals. We have also almost completely replaced the commercial software with home-built programs, which we discuss in Section 2.5. Our AFM itself (MultiMode, from Digital Instruments) also has some minor modifications which we discuss in Section 2.4.

2.2 Procedure

The proteins we unfold are polymeric repeats of the protein domain we want to study. Titin is one of the largest known proteins, and its structure varies between species and muscle types [8]. In order to have tractable datasets and broadly applicable results, we restrict our study to a single immunoglobulin-like domain (I27), which occurs several times in all native Titins [9]. In order to separate our signal unfoldings from noise and indeterminate surface interactions, we do not unfold single I27 domains, but a synthetic protein consisting of a number of consecutive I27 domains. The actual number of repeats is not critical, but the more repeats you have, the clearer the unfolding signature. The current project uses twelve consecutive I27 domains.

Our synthesized proteins also have a few cysteine amino acids on one end, which datively bond to gold. We deposit 10μL of protein solution (100μg/mL) onto a fresh gold surface, wait several minutes to allow the protein to bond to the gold, fill the AFM fluid cell with buffer solution, and then use the AFM tip to pull the free end of the protein away from the surface, unfolding the protein as the tip retreats.

The procedure for obtaining a force curve is as follows (see the schematic in Figure 4a):

1. Approach the AFM tip to the surface until the tip bends back a set amount.
2. Wait $\sim$10 seconds for the protein to bind to the AFM tip (The binding mechanism is poorly understood).
3. Withdraw the tip at a constant speed while measuring the cantilever deflection, until the protein breaks away from the tip.
4. Move to a different location on the sample.
5. Repeat steps 1-4.

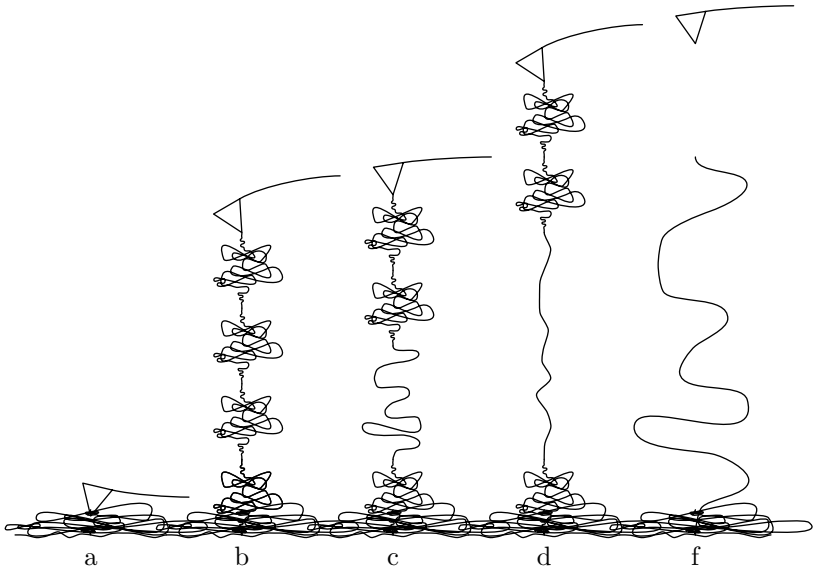
This procedure yields curves similar to that shown in Figure 4b. Most of the data we acquire is not as clean as this curve. We discuss how we deal with the complications in Section 2.6.

The shape of the rising edge of the sawtooth (c to d in Figure 4), can be modeled as a *worm-like chain* (WLC)[10]. The WLC model approximates the protein as an entropic spring, which exerts a force

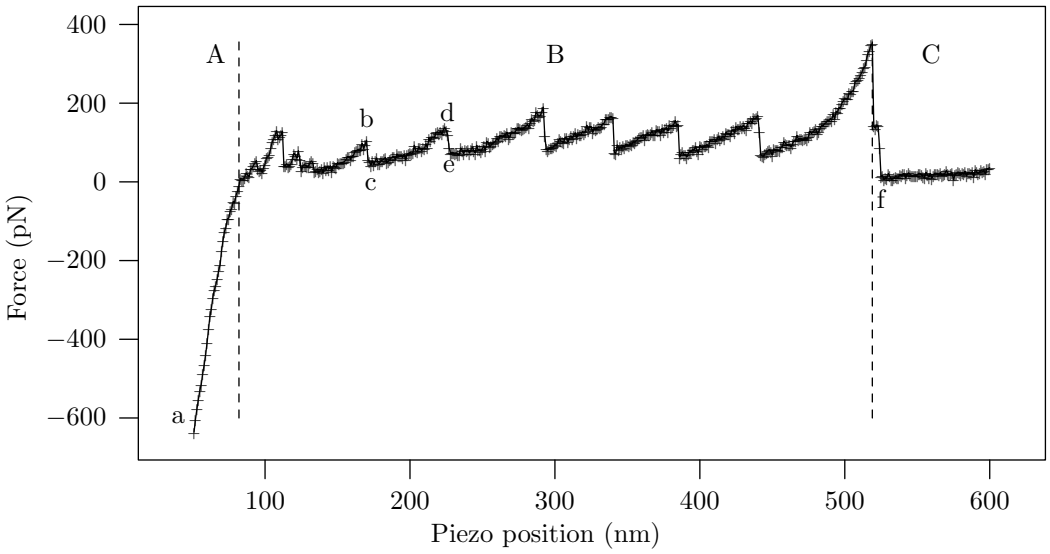
$$F(x) = \frac{k_B T}{p}\left[\frac{1}{4}\left(\frac{1}{(1-x/L)^2} - 1\right) + \frac{x}{L}\right], \tag{1}$$

where x is the distance between the pinned ends of the protein, p is the persistence length, and L the contour length of the protein chain.

Fitting our sawtooth to a succession of WLCs with different contour lengths yields the contour length increase upon a domain unfolding, which should be consistent with the value calculated from the domain structure. Average values of unfolding force as a function of temperature and pulling speed provide the data for computing the energy landscape roughness using Hyeon and Thirumalai's method[3], which we discuss in Section 2.7.



(a) The system.



(b) An example force curve.

Figure 4: Force curve explanation: The tip is initially pressed into the surface at point a with a predetermined force. In region A, the AFM tip retreats but is still in contact with the surface, and the force decreases as the enforced displacement of the tip decreases (the positive force direction in 4b is towards the surface). In region B the protein unfolds. At point b, the tip has retreated enough to pull the protein tight, giving the first force peak. One protein domain unfolds, relieving the strain on the protein chain and returning the deflection to near-zero at c. The retreating tip pulls the new conformation tighter until another domain unfolds from d to e. The process continues until there are no domains left to unfold, and the protein breaks off of the tip at f, leaving the tip in region C with no deflection.

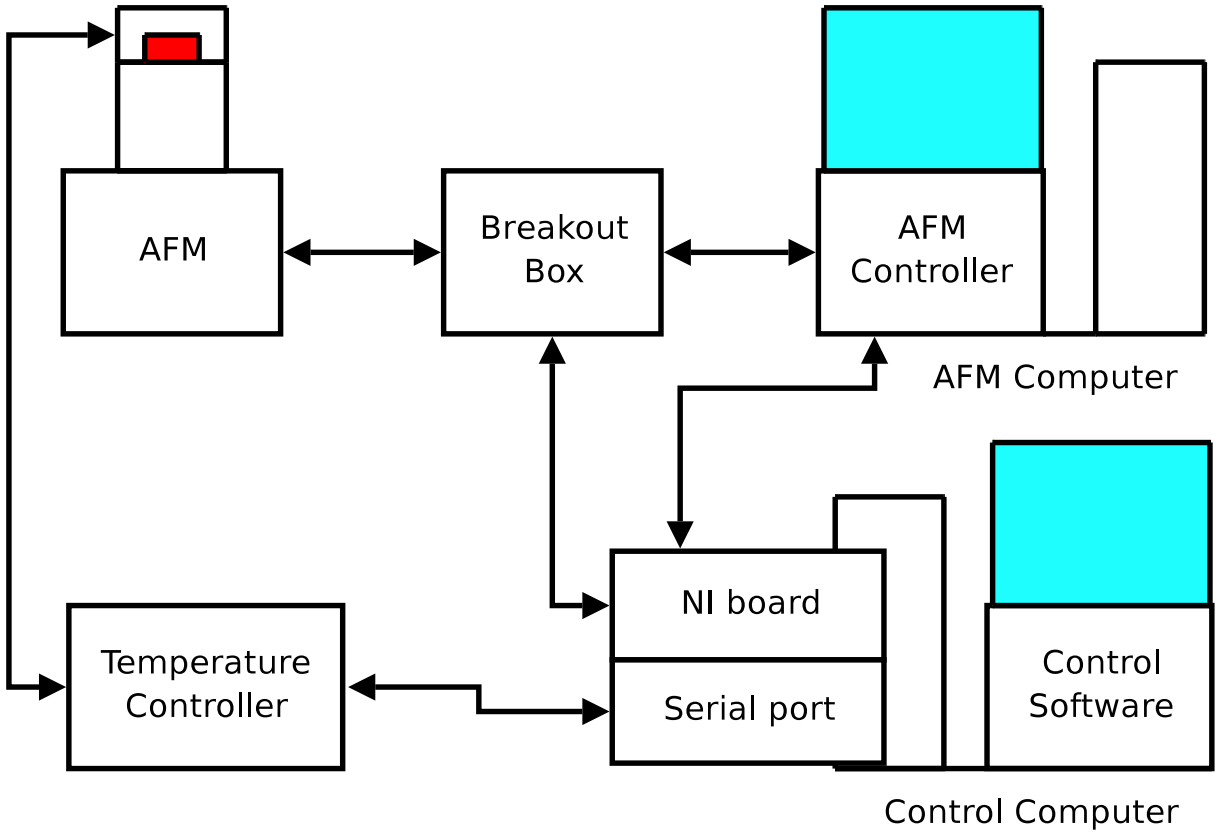


Figure 5: Equipment layout. The AFM contains the actual test system, as well as some simple processing electronics. The AFM controller sources the high voltages needed to drive the AFM piezo scanner. The temperature controller monitors and maintains a set temperature in the AFM fluid cell. Our software controls the whole setup, using a serial port to talk to the temperature controller, and an analog/digital output/input board to run the rest. The breakout box allows direct access to those lines of the AFM↔controller cable whose roles we understand and want direct access to.

Most of the actual equipment is contained in the commercial AFM. We have added a temperature control device consisting of a Peltier temperature module powered by a commercial temperature controller as discussed in Section 2.3. The remainder of our setup consists of controlling and monitoring electronics, discussed in Sections 2.4 and 2.5. A schematic of the equipment used in our experiment is given in Figure 5.

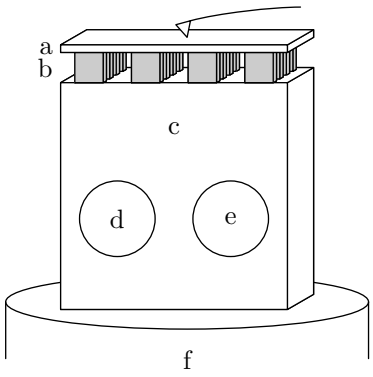
2.3 Temperature control

A temperature control apparatus for the AFM should be capable of maintaining biologically relevant temperatures (0°C to 50°C). For most proteins, this range is sufficient to measure the energy landscape roughness. The controller must not introduce much mechanical noise or thermal drift into the system. With our commercial AFM, we have the added constraint that the temperature module must have a small size to fit inside the MultiMode head.

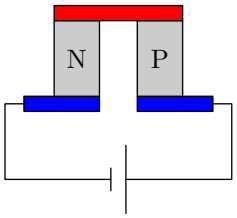
A temperature module satisfying these conditions was constructed around a Peltier thermoelectric heater/cooler (see Figure 6a). A Peltier functions by applying a voltage to regions of P- and N-type semiconductor in series (Fig. 6b). Applying a positive voltage as shown in Figure 6b heats the sample. Reversing the applied voltage cools the surface. A commercial thermoelectric controller (model MTCA 6040, from Melcor, Trenton, NJ) monitors the sample temperature with a thermocouple and powers the Peltier as required to maintain a set temperature.

The characteristics of the completed device are very encouraging[11]. Constant sample temperatures can be maintained over the entire cooling range as shown in Figure 7a. The system also responds relatively quickly to even large changes in temperature setpoint as demonstrated in Figure 7b. In order to set an upper limit on the vibrational noise due to the addition of the temperature control module, we imaged sputtered gold at our cold temperature extreme of 5°C and also under normal conditions at 25°C. As shown in Figure 7c, we see nanometer sized features in both images, which puts an acceptable upper bound on noise in the entire apparatus.

In conclusion, the temperature control system satisfies all the requirements necessary to allow temperature dependent unfolding experiments. In Section 2.8 we present some Ubiquitin unfolding results taken with this system.

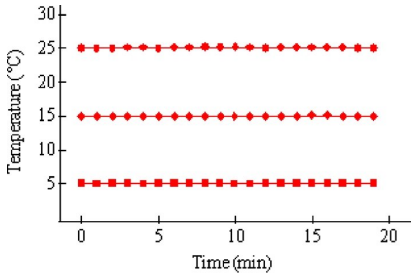


(a) Control module.

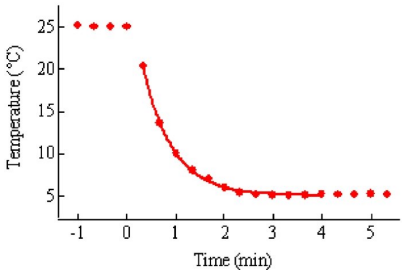


(b) Peltier.

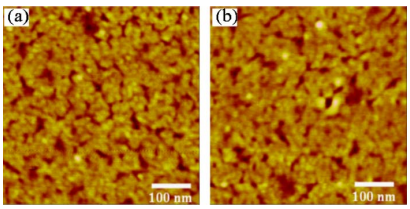
Figure 6: (a) Our temperature control module. The surface a is cooled or heated by one side of the Peltier thermoelectric device b. The other side of the Peltier is kept near room temperature using the brass heat sink c. The heat sink itself can be anchored to room temperature by gravity-fed water flowing through a U-shaped channel d to e. The module is mounted directly on our piezo scanner f. (b) A Peltier operating in heating mode. Regions of P- and N-type semiconductor connect hot and cold conducting plates.



(a) Temperature stability.



(b) Temperature response.



(c) Resolution.

Figure 7: Performance of the temperature control system[11]. (a) The stability of the sample temperature for setpoints of 5, 15, and 25°C. The temperature stabilized at 5.1, 14.9, and 25.1°C respectively, with fluctuations of less than 0.1°C. (b) The rate of temperature response to a setpoint change from 25 to 5°C. (c) Images of sputtered gold at (c.a) 5°C and (c.b) 25°C. The images are from different regions of the sample and were acquired in pure water in the tapping mode. Both images have a size of 500 x 500 nm^2.

2.4 Stepper control

Because of thermal drift and mechanical instability, the distance between the tip and the surface changes significantly over time. When the distance change exceeds the range of the piezo scanner, the stepper motor must be engaged to reposition the AFM tip relative to the sample. The previous software lacked the ability to control the motor on its own, so it would pause roughly every half hour and prompt the operator to make the necessary manual adjustments. Automatic motor control would allow the system to run longer without interrupts, facilitating the collection of large data sets. As a side benefit, automatic motor control would make equipment setup easier, as getting the tip close enough to seal the fluid cell o-ring without breaking the tip off on the surface is difficult for novices. Although automatic control of a stepper motor is a fairly simple task, difficulties in integrating the new control with the old experiment software led to the creation of the experiment server project discussed in Section 2.5 and an increased understanding of the inner workings of our AFM.

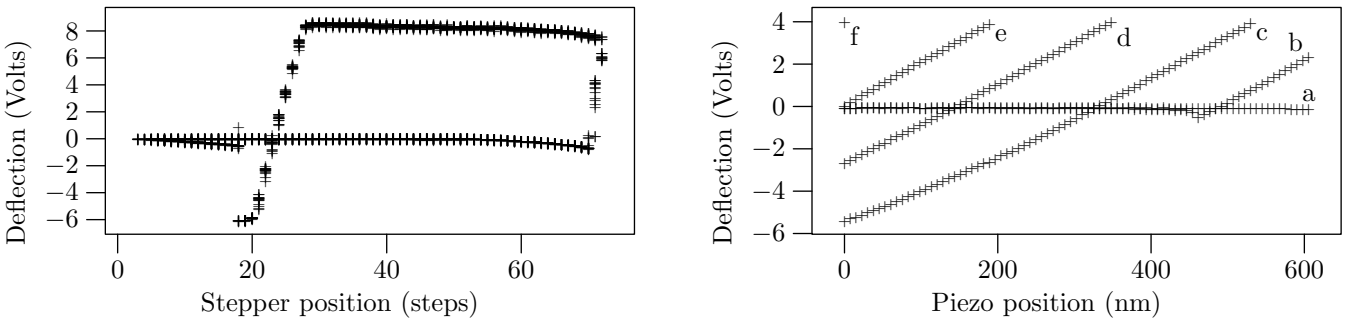
The first step in controlling the motor was to understand the way it was currently being controlled. Our AFM is a MultiMode II with a NanoScope IIIa controller. The MultiMode contains the piezo and motor for positioning the surface, as well as a circuit board handling some portion of the control logic and signal processing.

The motor can be controlled either by a switch on the body of the MultiMode, or programmatically via the NanoScope. Because there was NanoScope control of the motor, the easiest entry point seemed to be the DB-25 cable connecting the MultiMode to the NanoScope (see Figure 5). The task was to identify which signals to send down which lines in the connecting cable.

By monitoring any likely control lines with an oscilloscope while controlling the motor from the NanoScope, we were able to identify the four motor lines carrying rectangular wave signals. This made it likely that the NanoScope control lines carried the information about whether or not to pass current through each of the four motor coils. We built a break-out box to gain access to the relevant lines, wrote a simple test program to move the motor in one direction[12], and succeeded in visibly moving the motor.

We verified the stability and reproducibility of the microscopic movement of the motor by making several approach-retreat cycles from the surface of $\sim$70 steps which resulted in the data displayed in Figure 8a. We also measured the distance the surface moved with every step by determining the change in deflection voltage as a function of peizo position as we stepped the AFM tip closer the the surface. Our stepsize data is displayed in Figure 8b.

The motor is very consistent when approaching the surface, which indicates that our control software is operating correctly. However, the motor exhibits some hysteretic behavior on a scale of $\sim$46 steps, which is almost certainly due to *backlash*, or slack in the motor-surface coupling machinery. The first 46 steps in a new direction take the slack out of the coupling, and further steps move the tip relative to the surface. The problem can be avoided entirely by simply replacing



(a) Hysteresis.

(b) Stepsize.

Figure 8: (a) Stepper motor reproducibility, stability, and backlash. The data are from a single continuous counterclockwise trace of 14 approach-retreat cycles. The jump from about (18, -6) to (17,-0.4) is the snap-off effect, where short-range attractive interactions between the tip and the sample require the tip to be actively pulled off surface. Signal noise is comparable to that expected by drift. (b) Motor step size calibration. The stepper gradually stepped closer to the surface, feeling forward with the piezo after each step. Successive motor positions yield traces a, b, c, d, e, and f. As the motor moves the sample closer, less piezo movement is required to approach to same deflection level. The average spacing between the traces is roughly 170nm. Traces c and d have regions of negative deflection because the tip no longer retracts far enough from the surface to break free of the snap-off effect. There is no backlash because the data were taken during a single approach.

"backwards motion by one step" with "backwards motion by 60 steps and forward motion by 59 steps".

One issue raised by backlash is that it might be the source of some of our surface drift, as the drive-chain relaxes towards some central value and pulls the surface with it. By oscillating into our eventual position, we could perhaps settle the system at the beginning, reducing the need for adjustments later on. While this is not a problem for the current unfolding experiments, it could be an issue for longer unfolding-refolding experiments.

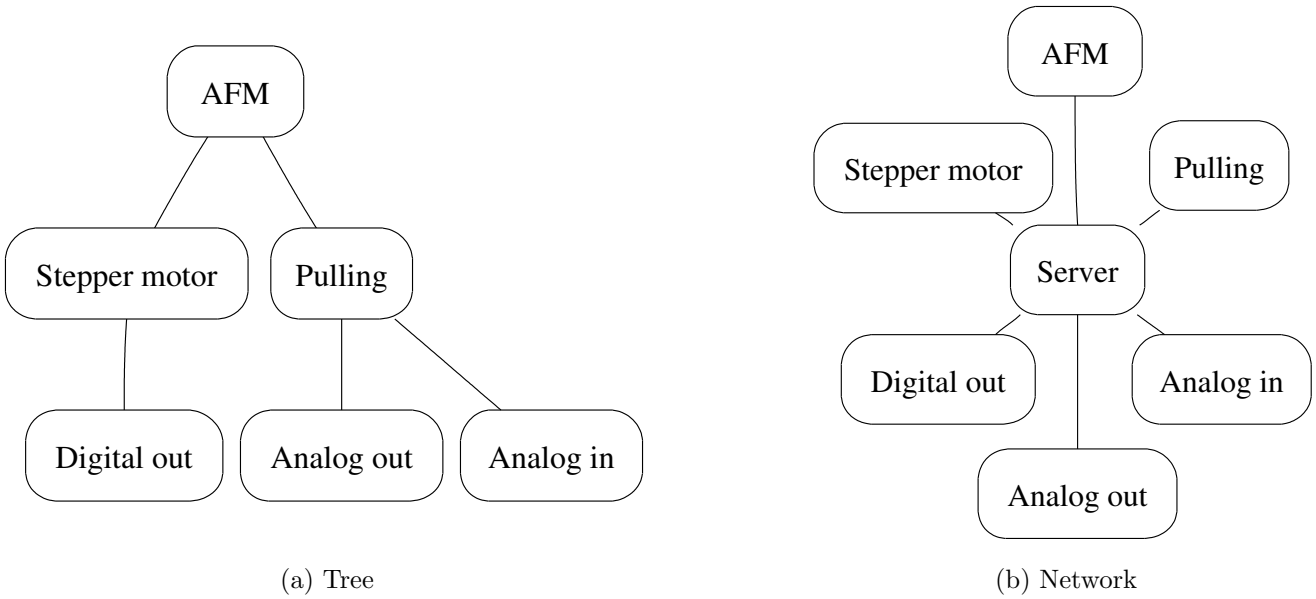
Having achieved motor control, the next order of business was to automatically approach the surface at the beginning of the experiment. As we already had access to the AFM's cantilever deflection signal, it was little trouble to implement our first algorithm, which simply stepped closer to the surface until some deflection set-point was acquired.

The first run of this program broke the tip off smashing into the surface. We realized that when the cantilever entered the bubble of liquid on the surface, the extra diffraction was deflecting the laser away from the photodiode. The photodiode still reported almost no deflection, but the reports were not valid. We needed to measure the total photodiode voltage as well, in order to differentiate the valid deflection signals from the invalid.

We expected the total photodiode voltage to also pass through one of the lines on the cable connecting the MultiMode to the NanoScope but after much searching were unable to determine

which line carried the signal. The total photodiode voltage is displayed on the MultiMode LCD screen, so we changed tactics and tried to determine which line connecting the MultiMode main board to the LCD screen carried the signal. After isolating the relevant line, we build a splitter cable to send the total photodiode voltage out through an auxiliary line in the MultiMode-NanoScope connector cable, and tapped into the auxiliary line through our break-out box.

We extended our initial approach algorithm to only trust validated deflection signals (see Appendix A for an implementation of the adjusted algorithm), and the new algorithm successfully approached the surface, or stops and signals the user that the photodiode has lost the signal. The process is not completely automatic, as the laser alignment knobs in the AFM head can only be adjusted manually, but laser re-alignment is only an issue during the initial setup and in cases of extreme drift, so this restriction should not be overly limiting.



(a) Tree

(b) Network

Figure 9: A comparison of (a) tree and (b) network software architectures. While the tree architecture favors structure and organization, the network architecture favors flexibility.

2.5 Experiment server

One source of rigidity in our previous software was its hierarchical tree architecture, with a single block of code orchestrating the whole proceedings (see Figure 9a). This setup allows well organized control over the experiment, but also imposes two restrictions. First, any functionality that requires user-set parameters must have those parameters passed to it down the chain of command. Therefore, if a low-level function needs more information, a method for passing the additional parameters down must be added to every function between the altered and root functions. Second, the experimenter needs to predict and prepare for many possible situations before the experiment actually begins. Many problems aren't recognized until the experiment is actually underway, and restarting the experiment after extending the software to cope with the errors is a time consuming process.

We decided to replace our tree architecture with a network architecture based on a central experiment server (see Figure 9b). Besides simplifying inter-process communication, the server would allow hot-pluggable processes, simple logging and debugging, and easily maintained client state information. Individual processes could be taken offline, adjusted, and reattached to the server without any change in real-world output. Diagnostic processes could also take over control of the real-world interface of a temporarily paused experiment processes to assist troubleshooting. An example of what the experiment-server could look like in operation is given in Appendix B.

The network uses a request-response message passing paradigm where a node that wishes to initiate an action or retrieve some information sends a request to another node and then waits for a response. To get an idea of the type of behavior we expect the server to handle, consider the

From	To	Message
Cmmdline	Stepper	Set:AbsPos 3
Stepper	Dig. Out	Set:Output 1
Dig. Out	Stepper	DONE
Stepper	Dig. Out	Set:Output 2
Dig. Out	Stepper	DONE
Stepper	Dig. Out	Set:Output 3
Dig. Out	Stepper	DONE
Stepper	Cmmdline	DONE

Table 1: Tracing the messages for a single command.

example conversation in Table 1.

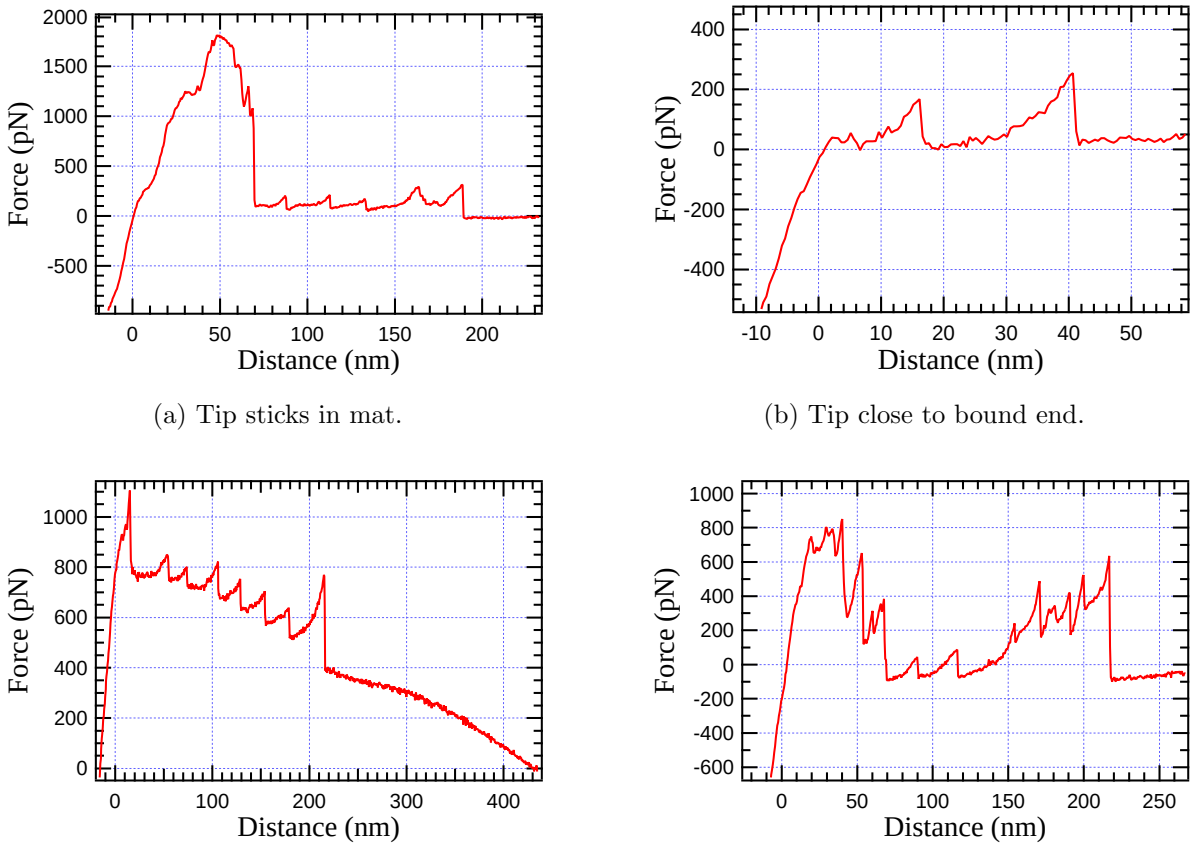
There are two paths towards adding functionality to the experiment control software: full fledged, experiment server clients and stand-alone, command line programs.

Many simple procedures are best implemented as stand-alone, command line programs. These programs can call server clients through the `Cmmdline` client, which allows easy text access to the server. Stand alone programs can in turn be called by server clients through the `Exec` client, which simply executes the command passed to it in the `bash` shell and returns the results. An example stand alone script for automatically approaching the sample surface is given in Appendix A.

More complicated procedures requiring preserved state information, complicated initialization, user feedback, or continuous monitoring are best implemented as full-fledged clients, so they can plug into the server and remain running for the duration of the experiment. The clients can be written in any language capable of handling sockets or linking to our library of socket routines, so the user can chose whichever language seems most suited to the task at hand.

The enforced modularization of code will allow easy testing of individual modules and let new users learn the software in manageable chunks. The hot-pluggable client-server architecture will allow diagnostic tools to be loaded on the fly. The common text interface will encourage the construction and reuse of flexible analysis and diagnostic tools, providing easier troubleshooting, as well as easing the transition between different software versions. Finally, simple logging by saving all messages passing through the server will increase the volume and quality of record-keeping, allowing easier forensic analysis of experiments-gone-wrong and providing sufficient material to reproduce interesting effects.

The experiment server is currently stable under light loads, and has been used to successfully run the surface approach script given in Appendix A.



(a) Tip sticks in mat.

(b) Tip close to bound end.

(c) Long range forces.

(d) Tip sticks to several proteins.

Figure 10: Classes of unfolding force curves. Note that each plot is scaled differently to highlight its distinguishing features.

2.6 Automatically sorting pulling curves

We attempt to unfold a protein by retracting the AFM tip from the surface (Sec. 2.2). However, most data does not look like that in Figure 4b. There are several possible outcomes when the AFM tip is pulled away from the surface. They are (roughly in order of their probability):

1. The tip does not stick to anything, and retreats without much deflection.
2. The tip sticks to a protein that's tangled in the mat of other proteins covering the surface.
3. The tip sticks to a protein very close to the end bound to the surface.
4. The tip sticks to a single protein close to the free end, and gives a clean unfolding force curve.
5. The tip sticks to several proteins, and unfolds them all in parallel.
6. Unidentified long range forces skew the measured force.

Example curves illustrating the unwanted behaviors are given in Figure 10.

The current procedure is to manually go through the data and pick out the 'good' curves by eye. We would like this process to be at least partially replaced by automatic sorting, which would clarify

%	Good	Bad
+	1.9	23
−	0.3	75

(a) Training set.

%	Good	Bad
+	2.2	3.9
−	2.2	91.5

(b) Test set.

Table 2: Comparison of manual and automatic sorting. (a) The training set of 620 AFM tip retractions. (b) The test set of 712 retractions. The rows represent the algorithms choices, and the columns represent manual choices. The parameters for each experiment were nominally identical, so the large variation between the two datasets is disturbing, but the algorithm is at least correlated with manual decisions.

sorting criteria, ease sorting of large data sets, and allow automatic tuning of experiment parameters to maximize yield. My current algorithm for sorting pulling curves is discussed in Appendix C.

After training the automatic sorter to mostly agree with our intuitive sorting on one data set, we tested the sorter by comparing it's results to our manual selection on another dataset. Our results are given in Table 2.

After a manual examination of the results, the automatic program did a good job of removing curves with long range forces or no unfolding forces, but had trouble separating the curves with clean protein unfolding sawtooth from curves with more complicated features. This difficulty is understandable, because the current algorithm only looks for steep drops in force with roughly the proper spacing, but does not fit the region between drops to the WLC model. Adding a WLC fitting step should be much more accurate, and will hopefully supplant manual sorting completely.

We do not yet understand the reasons behind the large difference between Tables 2a and 2b. There may be some variables we are not controlling, or it may simply be spatial irregularity in the manner the protein is distributed over the sample. Once the automatic sorter becomes more reliable, we hope to clarify the reasons behind this variation.

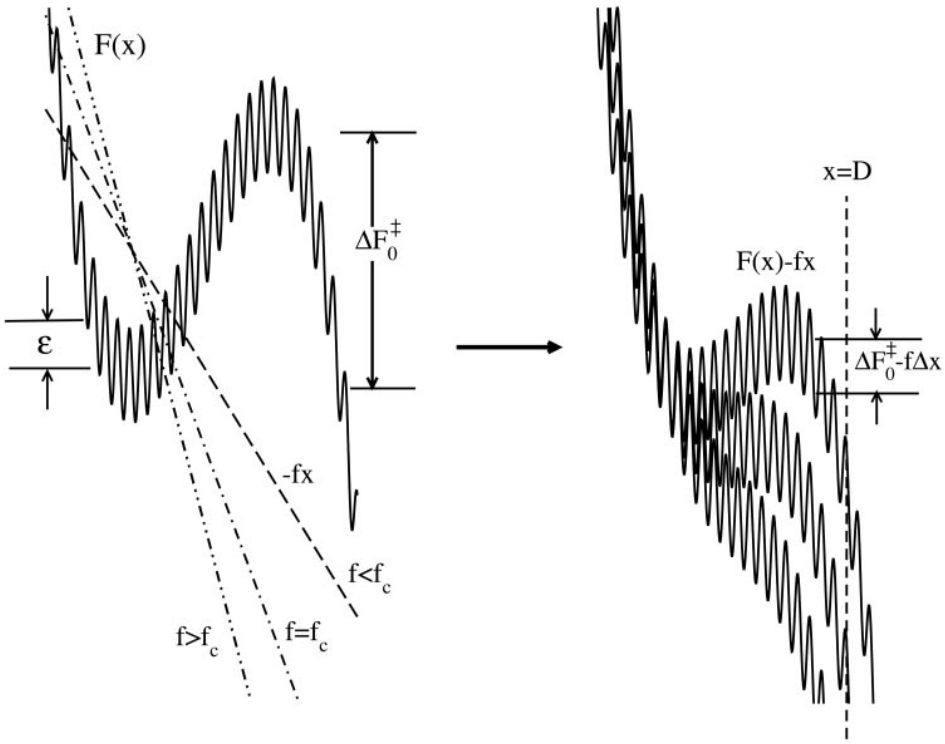


Figure 11: Proposed energy roughness model[3]. The free energy $F(x)$ is given by $F(x) = F_0(x) + F_1(x)$, where $F_0(x)$ is the smooth free energy funnel, and $F_1(x) \sim \varepsilon$ is the local roughness in free energy.

2.7 Measuring energy landscape roughness of protein molecules

Kramers' theory is the study of transition rates between meta-stable energy wells[13]. Hyeon and Thirmulai combine Zwanzig's work on diffusion in frustrated potentials to two state systems and Kramers' theory to develop a theory of how frustration affects escape from one well into the other[3][14]. Hyeon and Thirmulai apply their theory to two-state protein unfolding, and show that measuring how the unfolding behavior varies with temperature yields information about the frustration energy scale.

Many proteins have funnel-shaped energy landscapes with limited frustration (Fig. 1). Let ε be the landscpe roughness energy scale. For proteins whose frustration is on the order of the thermal energy in the environment ($\varepsilon \sim k_B T$), the protein is never caught in any particular local minimum, but wanders in the vicinity of its global minimum (see Figure 11). For such minimally frustrated proteins, the two-state model describes the protein's conformation as either generally 'folded' or generally 'unfolded' without worrying about the details added by many local minima.

In our mechanical unfolding experiments, we have a well defined reaction coordinate in the protein's end-to-end distance x. For Titin, which is naturally tethered at both ends, this reaction coordinate has the added benefit of direct biological significance. The well to the left in Figure 11 represents our originally stable folded states, and the well to the right the unfolded states. As

we apply an unfolding force to the potential, we lower the free energy of the transition states, encouraging unfolding. The frustration $F_1(x)$ servers to limit the diffusion of proteins on this landscape, increasing the force or time necessary to unfold the protein. Hyeon and Thirmulai use Zwanzig's method to find the effective potential $F^*(x)$ under a constant external force f is given by

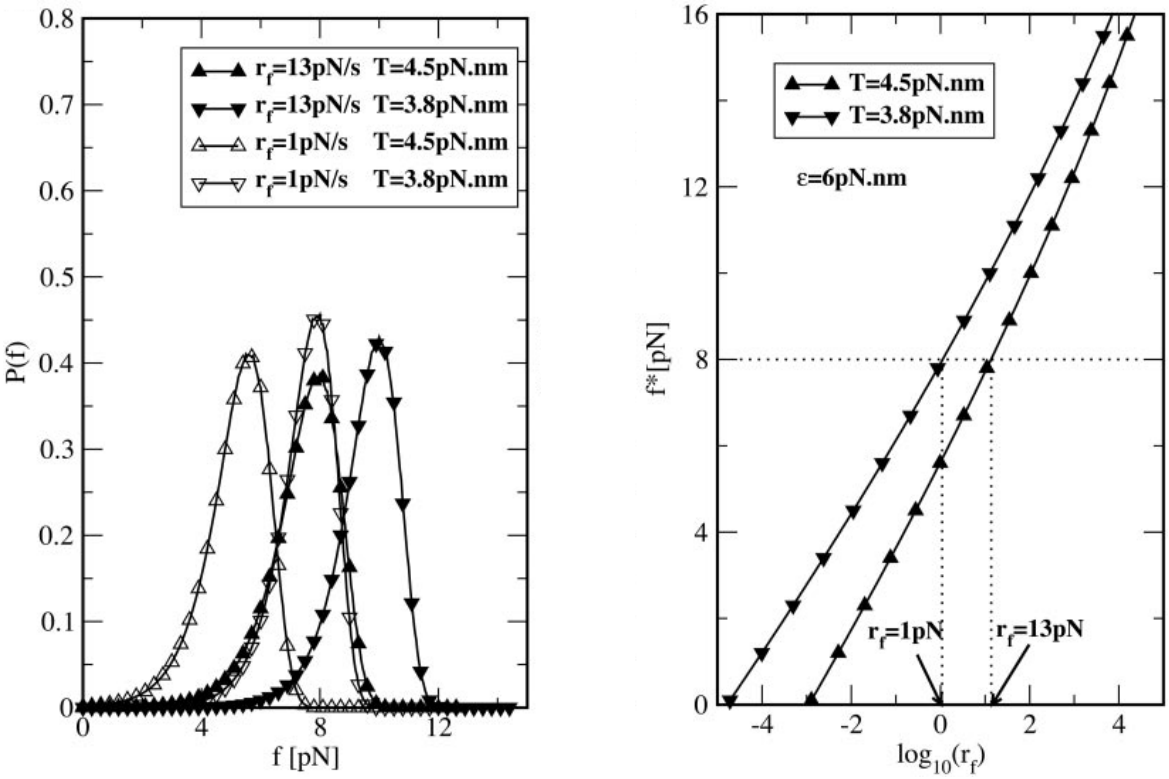
$$F^*(x) = F_0(x) - fx - k_B T \psi^- \tag{2}$$

Where the first term is the frustration-less funnel potential, the second term is the potential due to an external force, and the third term is a temperature dependent 'diffusion force', with $\psi^- \equiv \ln(\langle e^{-\beta F_1(x)} \rangle)$, $\beta \equiv 1/k_B T$, and $\langle \ldots \rangle$ denoting the local average of the enclosed quantity.

Applying Kramers' theory to this potential for experiments with constant force loading rates $r \equiv df/dt$, they predict how the most common unfolding force f^* will vary with temperature, loading rate, and energy landscape roughness ε. For two different temperatures T_1 and T_2, measuring the loading rates $r(T_1)$ and $r(T_2)$ required to produce the same f^* will yield ε using

$$\varepsilon^2 \approx \frac{k_b^2 T_1 T_2}{T_2 - T_1} \cdot \left[T_1 \log\left(\frac{r_f(T_1)\Delta x(f^*)}{k_0(f^*) k_B T_1} \right) - T_2 \log\left(\frac{r_f(T_2)\Delta x(f^*)}{k_0(f^*) k_B T_2} \right) \right], \qquad \text{Hyeon Eqn. 9}$$

The simulated data supporting this formula is given in Figure 12. Our experiment replicates this simulation experimentally to extract ε for our selected protein.



(a) Unfolding distributions.

(b) Most common unfolding force.

Figure 12: Simulated constant loading experiment for determining ε[3]. (a) Four unfolding force probability distributions. The most common unfolding force f^* from each distribution in (a) becomes a point in (b), which shows the dependence of f^* on the loading rate for two temperatures.

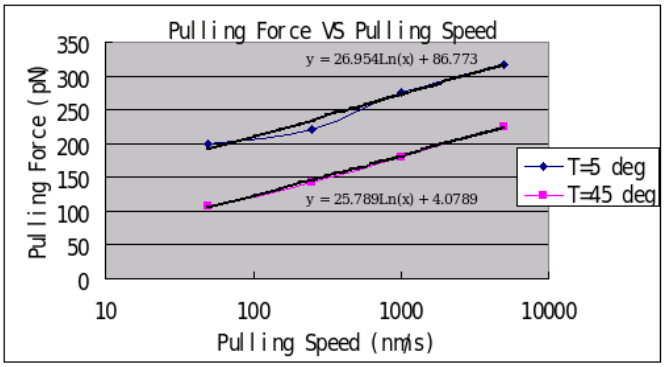


Figure 13: Temperature and loading rate dependence of unfolding in Ubiquitin[11]. This figure is the experimental equivalent to Hyeon and Thirmulai's simulated data in Figure 12.

2.8 Energy landscape frustration measurements in Ubiquitin

As a test of the temperature control system, Y. Yang et al. applied the method of Hyeon and Thirmulai to calculate the frustration energy scale in octomeric Ubiquitin[3][11], measuring the most probable unfolding force as a function of loading rate and temperature. Their data is presented in Figure 13. When the most likely unfolding force is 200pN, the pulling speed is $v_1 = 66.7$nm/s at $T_1 = 5°$C and $v_2 = 1992$nm/s at $T_2 = 45°$C. With a spring constant of $\kappa = 85.6$pN/nm measured by thermal vibration, the loading rates are then given by $r_f(T_1) = \kappa v_1 = 5648$pN/s and $r_f(T_2) = \kappa v_2 = 1.69 \cdot 10^5$pN/s. Combined with the values from the literature of $k_0 = 5.0 \cdot 10^{-5}$Hz and $\Delta x = 0.225$ nm [15], we can calculate the unfolding energy according to Hyeon and Thirmulai's Equation 9, and extract $\varepsilon \approx 1.98 \cdot 10^{-20}\text{J} \approx 4.8 k_B T$. Because $\varepsilon > k_B T$, this frustration is in the upper limit of frustrations for which Hyeon and Thimulai's formulation is valid.

This measurement is the first experimental measure of a protein energy landscape frustration scale, although the same technique has been used to measure the frustration of receptor-ligand binding [16]. The difficulty with this experiment is a combination of the extra equipment needed to control the temperature and the large datasets required to extract ε. Our streamlined system should greatly facilitate making such measurements on Titin.

3 Conclusions

We attempt to measure the free energy landscape roughness of protein molecules and to gain insight into the molecular mechanisms behind temperature changes in muscle stiffness. To facilitate these measurements, software and hardware improvements to the experimental setup have been designed, constructed, and tested. Specifically, the stepper motor has been automated, experiment server software has been written, and an automatic sorting algorithm is in development.

In the future, we hope to measure the frustration scale of the Titin domain I27, which will also server to stress-test the experiment server and sorting algorithm. We also hope to add WLC fitting to our automatic curve sorter, removing as much subjectivity from our analysis as possible. We will make a systematic study of the unfolding and refolding behaviors of I27 within the temperature range from 5°C to 50°C. The results will be analyzed using the theory of Hyeon and Thirmulai. Once the Titin measurements have been completed, our procedure should be stable enough for easy application to other proteins of interest, adding valuable information about the basic physics behind the protein folding problem.

Appendices

A An example client script

```
#!/bin/sh
TARGET_DEF=1 # Volts
MIN_TPHOTO=3 # Volts
DEF=0 ; TPHOTO=0 ; ANS=0 ;
get_feedback() {
  DEF=‘cmmdline -c ’:AFM|Get:Deflection’‘
  TPHOTO=‘cmmdline -c ’:AFM|Get:TotalPhotodiode’‘
}
step_closer() {
  ANS=‘cmmdline -c ’Stepper|Set:RelPos 1’‘
}
get_feedback
while [ $DEF -lt $TARGET_DEF ] && [ $TPHOTO -gt $MIN_TPHOTO ]; do
  step_closer
  get_feedback
done
```

B An example command-line experiment

This artist's representation should give a feel for how an experiment-server run experiment looks from the command line.

```
> approach_surface
Lost signal!
> approach_surface
> piezo_measure_distance
372 nm
> pull_protein --save 'D:\data\' &
saving data to 'D:\data\070510\'
> monitor_unfolding
> sort_data --dir 'D:\data\070510\'
...
> ls
pulls_good
pulls_bad_structure
pulls_no_unfolding
...
```

C Sorting criteria

The current procedure for sorting force curves is summarized in the flow chart in Figure 14. An explanation of what the criteria are and whee they come from follows:

- The last quarter of the data is fit to a straight line. If the fit line is steeper than a given threshold, the curve has forces as far out as we pulled, so it is thrown out. Otherwise, we use the average value at the end of the line to define zero force.
- Once we know what zero force is, we can pick out the upward deflected contact region from the beginning of the curve. We also fit this to a line.
- We slide down the curve from the zero force side until we find a downward deflection force above a given threshold. This marks snapping-off of the end of protein (or the surface, if there was no protein attachment).
- Once we've isolated the protein portion of the curve, we can check to see if it is approximately the right length, culling curves that are too long or too short.
- Now we look for peaks for possibly matching a clean sawtooth. We low pass the protein to reduce noise, and pick out large trough minimums. Going back to the non-lowpassed data, we find the point of most force between each pair of minimums.

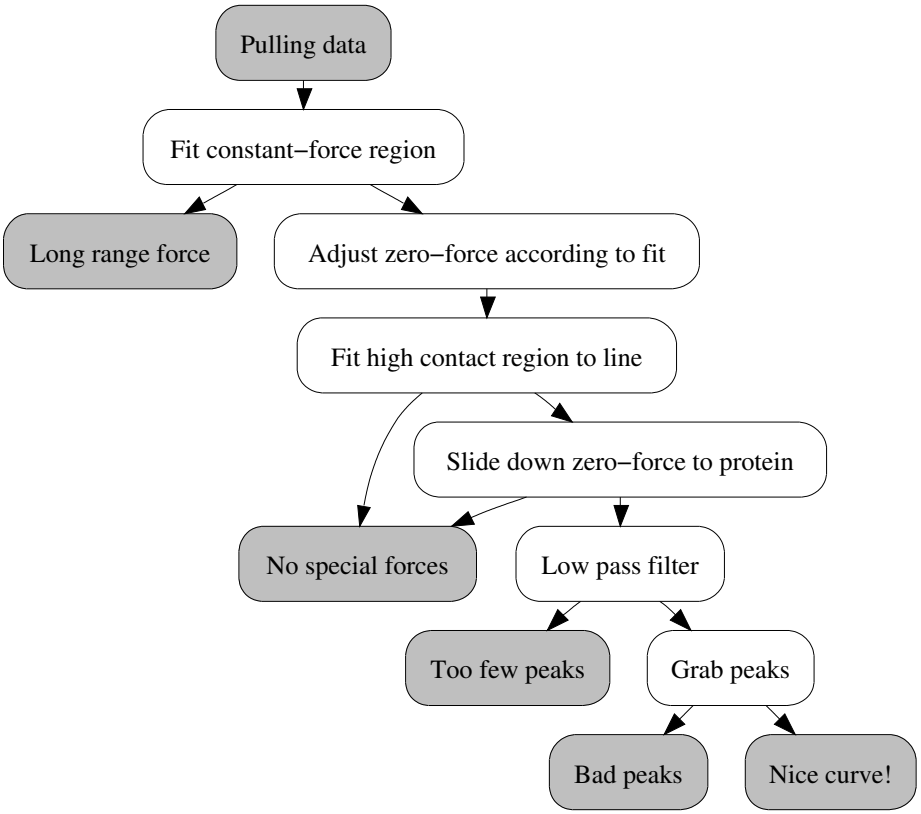


Figure 14: Automatic data sorting logic.

- We then check to see that the drop following the high force is deep and steep enough for a clean sawtooth. This is the unreliable node, which should be replaced by a WLC fit to improve selection efficiency.

References

[1] C. Levinthal. Mossbauer Spectroscopy in Biological Systems: Proceedings of a meeting held at Allerton House, Monticello, Illinois. Editors: J.T.P. DeBrunnerand E. Munck. University of Illinois Press (1969), 22-24.

[2] M. Carrion-Vazquez, A.F. Oberhauser, S.B. Fowler, P.E. Marszalek, S.E. Broedel, J. Clark, and J.M. Fernandez. *Mechanical and chemical folding of a single protein: A comparison.* Proc. Natl. Acad. Sci. USA (1999) **96**, 3694-3699.

[3] C. Hyeon and D. Thirumalai. *Can energy landscape roughness of proteins and RNA be measured by using mechanical unfolding experiments?* PNAS (2003) **100**, 18 10249-10253.

[4] R.S. Kirton, A.J. Taberner, P.M.F. Nielsen, A.A Young, and D.S. Loiselle. *Effects of BDM, $[Ca^{2+}]_o$, and temperature on the dyanmic stiffness of quiescent cardiac trabeculae from rat.* Am J Physiol Heart Circ Physiol (2005) **288**, 1662-1667.

[5] S. Saito, M. Sasai, and T. Yomo. *Evolution of the folding ability of proteins through functional selection.* PNAS Biophysics (1997) **94**, pp 11324-11328.

[6] M. Gautel. Kings College, London. Cardiovascular Division. Accessed May 31, 2007. `http://www.kcl.ac.uk/schools/medicine/research/cardio/pi/gatuel-m.html`

[7] G. Warr. *Homepage.* U. of Sydney, School of Chem. Accessed May 12, 2007. `http://www.chem.usyd.edu.au/~warr_g/AFM3.jpg`

[8] G. Mutungi and K.W. Ranatunga. *Tension Relaxation after Stretch in Resting Mammalian Muscle Fibers: Stretch Activation at Physiological Temperatures.* Biophysical Journal (1996) **70**, 1432-1438.

[9] D.O. Fürst and M. Gautel. *The anatomy of a molecular giant: How the sarcomere cytoskeleton is assembled from immunoglobulin superfamily molecules.* J. Mol. Cell. Cardiology (1995) **27**, 951-960.

[10] M. Carrion-Vazquez, A.F. Oberhauser, T.E. Fisher, P.E. Marszalek, H. Li, and J.M. Fernandez. *Mechanical design of proteins studied by single-molecule force spectroscopy and protein engineering.* Progress in Biophysics & Molecular Biology (2000) **74**, 63-91.

[11] Y. Yang, F. Lin, and G. Yang. *Temperature control device for single molecule measurements using the atomic force microscope.* Rev. Sci. Instrum. (2006) **77**, 063701.

[12] D. Jones. *Control of Stepping Motors.* Accessed May 19, 2007. `http://www.cs.uiowa.edu/~jones/step/`

[13] P. Hänngi, P. Talkner, and M. Borkovec. *Reaction-rate theory: fifty years after Kramers.* Rev. Mod. Phys. (1990) **62**, 2, 251-342.

[14] R. Zwanzig. *Diffusion in a Rough Potential.* Proc. Natl. Acad. Sci. USA **85** (1988), 2029-2030.

[15] C. Chyan, F. Lin, H. Peng, J. Yuan, C. Chang, S. Lin, and G. Yang. *Reversible Mechanical Unfolding of Single Ubiquitin Molecules.* Biophysical Journal (2004) **87**, 3995-4006.

[16] R. Nevo, V. Brumfeld, R. Kapon, P. Hinterdorfer, and Z. Reich. *Direct measurement of protein energy landscape roughness.* EMBO Rep. (2005) **6**, 482-486.