Measuring biomolecules: an image processing and length estimation pipeline using atomic force microscopy to measure DNA and RNA with high precision

Andrew Sundstrom

October 24, 2008

Abstract

Background. An important problem in molecular biology is to determine the complete transcription profile of a single cell, a snapshot that shows which genes are being expressed and to what degree. Seen in series as a movie, these snapshots would give direct, specific observation of the cell's regulation behavior. Taking a snapshot amounts to correctly classifying the cell's ∼300 000 mRNA molecules into ∼30 000 species, and keeping accurate count of each species. The cell's transcription profile may be affected by low abundances (1-5 copies) of certain mRNAs; thus, a sufficiently sensitive technique must be employed. A natural choice is to use atomic force microscopy (AFM) to perform single-molecule analysis. Reed *et al.* ("Single molecule transcription profiling with AFM.", *Nanotechnology*, 18:4, 2007) developed such an analysis that classifies each mRNA by first multiply cleaving its corresponding synthesized cDNA with a restriction enzyme, then constructing its classification label from ratios of the lengths of its resulting fragments. Thus, they showed the transcription profiling problem reduces to making high-precision measurements of cDNA backbone lengths — correct to within 20-25 bp (6-7.5 nm).

Contribution. We developed an image processing and length estimation pipeline using AFM that can achieve these measurement tolerances. In particular, we developed a biased length estimator using James-Stein shrinkage on trained coefficients of a simple linear regression model, a formulation that subsumes the models we studied.

Methods. First, AFM images were processed to extract molecular objects, skeletonize them, select proper backbone objects from the skeletons, then compute initial lengths of the backbones. Second, a linear regression model was trained on a subset of molecules of known length, namely their computed image feature quantities. Third, the model's coefficients underwent James-Stein shrinkage to create a biased estimator. Fourth, the trained and tuned model was applied to the image feature quantities computed for each test molecule, giving its final, corrected backbone length.

Results. Training data: one monodisperse set of cDNA molecules of theoretical length 75 nm. Test data: two monodisperse sets of cDNA molecules of unknown length. Corrected distributions of molecular backbone lengths were within 6-7.5 nm from the theoretical lengths of the unknowns, once revealed.

Conclusions. The results suggest our pipeline can be employed in the framework specified by Reed *et al.* to render single-molecule transcription profiles. The results reveal a high degree of systematic error in AFM measurements that suggests image processing alone is insufficient to achieve a much higher measurement accuracy.

1 Introduction

1.1 Motivation

An important problem in molecular biology is to determine the complete transcription profile of a single cell, a snapshot that shows which genes are being expressed and to what degree. Seen in series as a movie, these snapshots would give direct, specific observation of the cell's regulation behavior. Taking a snapshot amounts to correctly classifying the cell's ∼300 000 mRNA molecules into ∼30 000 species, and keeping accurate count of each species. The cell's transcription profile may be affected by low abundances (1-5 copies) of certain mRNAs; thus, a sufficiently sensitive technique must be employed. A natural choice is to use atomic force microscopy (AFM) to perform single-molecule analysis. Reed *et al.* [28] developed such an analysis that classifies each mRNA by first multiply cleaving its corresponding synthesized cDNA with a restriction enzyme, then constructing its classification label from ratios of the lengths of its resulting fragments. Thus, they showed the transcription profiling problem reduces to making high-precision measurements of cDNA backbone lengths — correct to within 20-25 bp (6-7.5 nm).

1.2 Problem statement

Given a high resolution ($< 1 \frac{nm}{pixel}$) AFM image produced under real experimental conditions (noisy, tip dilation and thermal drift effects present), containing arbitrary short DNA molecules (e.g., cDNAs, mitochondrial DNAs or short fragments of genomic DNAs), compute estimates of their backbone contour lengths that are accurate to within a specified tolerance, e.g., an estimation error with a standard deviation of 7.5 nm.

1.3 Related work

For more than a decade, researchers have investigated the problem of how to accurately measure DNA contour length by computer analysis of AFM images. This work falls into three broad categories: manual methods, where human operators hand-draw piecewise linear backbones over objects extracted from the image background[1]; semi-automated methods [26] that involve human interaction with image processing and object segmentation algorithms; and automated methods [33, 34, 11, 29, 30, 14, 16, 15, 13, 5] that perform their analysis and measurement unsupervised. For reasons of speed and reproducibility, we focused our investigation on automated methods.

The problem breaks down into two steps: image processing, then length estimation. Image processing takes as input an AFM image of high resolution (say, 1024×1024 pixels representing a microscopic area of 1000×1000 nm) and outputs a set of one-dimensional, eight-connected pixel paths in a transformed image that form the discrete representation of the continuous molecule backbone contours. Length estimation assigns to these backbones numerical values that purport to measure the true end-to-end length of the molecules.

All of the automated processing methods employ a pipeline of image processing steps. In common are steps that remove noise, extract foreground objects, iteratively erode each two-dimensional object into a joined one-dimensional line structure (tree), and finally prune each tree's branches from its trunk — the backbone contour to be measured next. The erosion (alternatively called *thinning* or *skeletonizing*) algorithms employed are surveyed in [24]. Some of the automated methods [33, 34, 14, 16, 15, 13] insert a step after erosion that uses a line-continuity heuristic to decide whether to recover tip pixels that were eliminated during the erosion step. One of the automated

[1] For example, by using a tool like NIH Image (http://rsbweb.nih.gov/nih-image/).

methods [5] innovates the last, tree-pruning step by transforming it from a strict image processing problem to a graph optimization one, where instead of eliminating branch pixels until the trunk is encountered, the tree is represented as a graph. In this scheme, a node is a pixel at the point of path bifurcation or path termination; an arc is a pixel path whose weight is given by a linear combination of two types of distance, determined by the relative orientations of consecutive pixel pairs: unit distance for horizontal and vertical, $\sqrt{2}$ for diagonal; the longest path traversal through this graph represents the trunk.

For nearly 50 years, since Freeman's pioneering work in the image analysis of chain-encoded planar curves [19], the study of contour digitization has received much attention. Namely, what is the most accurate estimator of the end-to-end length of an arbitrary continuous contour that underlies its discrete representation as a one-dimensional pixel path? The literature contains numerous estimators, and frameworks to evaluate their relative performance [9, 10, 42, 25, 32, 23, 18, 6, 21]. All of the automated processing methods mentioned above employ a pipeline of length estimation steps chosen from this set of estimators. These pipelines' approaches vary, from those that simply traverse the chain-code to yield a linear combination of unit and $\sqrt{2}$ distances [33, 34, 11, 5], to those that use one of a variety of parametric estimators [29, 14, 16, 15, 13], to one that takes a signal processing approach based on fast-Fourier transformation followed by Gaussian filtering and normalization [30].

A related focus of investigation involves estimating the *intrinsic curvature* of DNA from AFM images [45, 17]. Intrinsic curvature of DNA is a function of the nucleotide sequence, independent of dynamic components of curvature brought on by thermal agitation. This work may eventually improve DNA backbone contour length estimates by inputting accurate estimates of curvature to a length estimator that models the DNA contour as a sequence of straight lines and circular arcs [42, 32, 21].

1.4 Our contribution

Our software system uses the image processing pipeline of Cirrone [5] and extends the length estimation pipeline in a novel way. First, we fit each backbone pixel path with a sequence of cubic splines, one for each five-pixel subpath, where the last pixel of a given subpath is the first pixel of the next (i.e. all subpaths share one extremity pixel). A tailing subpath, $\mathcal{T}$, having $p < 5$ pixels is handled by fitting a cubic spline to the subpath formed by prepending to $\mathcal{T}$ the prior $5 - p$ pixels, then counting the spline's length from its closest approach to the first and last pixels in $\mathcal{T}$. The resulting summed length of the cubic splines gives the initial backbone length estimate, L_{CS}.

We correct L_{CS} by a linear combination of various features, some examined in the literature, some of our own design, such as: number of horizontal pixel pairs, number of vertical pixel pairs, number of diagonal pixel pairs, number of corner pixel triplets, mean backbone intensity value (height in an AFM image analysis setting), standard deviation of backbone intensity value, mean backbone thickness measured at each pixel, and standard deviation of backbone thickness. The true length, $\mathcal{L}$, is thus modeled as L_{CS} plus a linear combination of the feature terms plus an error term, ε, where the feature term coefficients derive from an overdetermined system of linear equations obtained from a set of calibrating molecules of known length. We assume $\varepsilon \sim N(0, \sigma^2)$ represents a Gaussian noise, thus satisfying the Gauss-Markov condition.

We were less concerned with estimation bias because we trained on a large population of calibrating molecules. In terms of mean squared error (MSE), we hoped to improve our length estimates by trading off a small amount of bias for a substantial reduction in variance. To accomplish this, we applied two types of James-Stein shrinkage [20, 35] to the trained feature term coefficients: spherical (uniform) [31] and truncated [8] (selective) shrinkage, the latter tested over the admissible

range of degrees of freedom. Our results showed marginal improvements after shrinkage, perhaps indicating the absence of multicollinearity in our bundle of feature variables. Regardless, we believe our system implements a meta-approach to the problem of feature-based length estimation. Those features that are less informative to backbone length prediction will have more variance; consequently, their coefficients will shrink to zero in a biased fashion. Thus, any number of image-based features may be incorporated into our simple linear model in an easily extensible way, giving rise to backbone length estimates whose error is not necessarily constrained by geometric lower bounds in terms of, for example, pixel density [9, 10, 32] or multigrid convergence [23, 6]. In this way, our approach subsumes those length estimation formulations comprised in small, fixed sets of backbone chain code parameters cited above.

The allegory of the seven blind mice inspired our meta-approach; it is captured by the proverb "Knowing in part may make a fine tale, but wisdom comes from seeing the whole." [43]. Each image-based feature provides limited predictive power for backbone contour length. But integrated into a properly chosen model, with each feature contributing according to its demonstrated informativeness during training, in principle, the collective result should be superior to any rendered by strict subsets, provided there is no over-fitting. Moreover, outside of computational complexity considerations, there should be no bound on the number of features one applies to the problem. The more features, the more potential to drive down the error. Each will tell its tale.

1.5 Problem status

In our experiments on monodisperse unknown length fragments, we achieved backbone contour length estimates accurate within 7 nm.

What more can be done?

First, all the approaches under review, including ours, make use of half of the AFM data available. For each point (x, y) in the area under inspection, the AFM instrument in tapping mode takes two measurements: the displacement in the z-direction for *height* (the typical AFM "image"), and the change in oscillation frequency for *softness*. Second, none attempt to model tip convolution effects directly and appropriately deconvolve the image, though the problem is widely acknowledged [22, 38, 7, 37, 40] and algorithms designed precisely for this purpose exist [39]. Third, none attempt to model thermal drift directly and perform the appropriate deblurring of the image, though this problem too is widely acknowledged [4, 41, 27, 44] and an assortment of well-suited algorithms for this exist, namely the work of Carasso [2, 3]. Fourth, experimenters can use closed-loop scanning settings in their protocols, to reduce the effects of thermal drift by spending the majority of scan time on just the objects of interest. These last three are sources of systematic error that can, in principle, be removed, and should lead to more accurate length estimates. In addition, there are problems implicit to the chemistry, namely, it is not well understood how a three-dimensional DNA or RNA molecule adsorbs onto a substrate like mica, and under what conditions uniform binding to the surface occurs, let alone how to ensure this. We expect better models will emerge that will eventually lead to reduction in these kinds of experimental error.

In terms of learning feature values from the data, we would like to explore alternatives to linear regression. For instance, AdaBoost with decision stumps. In terms of biased estimation, we would like to explore alternatives to James-Stein shrinkage. For instance, principal component regression, ridge regression, and LASSO — subjects for a follow-up study.

2 Methods

Cirrone [5] implemented an application called *AFM Explorer*, using the *wxWidgets*[2] and *OpenCV*[3] libraries. It provides a graphical user interface (GUI) that allows the user to adjust image processing parameters (e.g. select from a set of intensity value thresholding methods and values), adjust the $\frac{nm}{pixel}$ image density factor, process an AFM image, and save the image at different steps of processing. Loading an AFM image places it in central view. Once the application runs the image through the image processing pipeline, it displays in separate tabbed views the skeletonized molecules and the final backbone contours, and in a separate area it lists the computed backbone contour lengths. The user can click on list entries to highlight the associated molecules in each image view, or vice-versa, allowing the user to establish a clear correspondence between visual and numerical results.

2.1 AFM Explorer image processing pipeline

AFM Explorer uses the image processing pipeline schematically presented in Figure 1. We outline the steps below (refer to [5] for full algorithmic detail).

The pipeline has three phases:

2.1.1 Filter

This is implemented as five calls to the *OpenCV* library. We begin with a 24-bit RGB image, presumably generated by the AFM apparatus image capture software. (Figure 2a). We first convert it into an 8-bit grayscale image (`cvCvtColor`), and then perform intensity level histogram equalization (`cvEqualizeHist`), to increase the local contrast in the image. We next smooth the image by setting the intensity level of a given pixel to the median intensity level of a 5×5 pixel window about it (`cvSmooth`). To create a binary image from the smoothed grayscale one, we first suppress pixels that have an intensity level below an empirically derived static threshold (`cvThreshold`). In a second pass, we adaptively promote to the maximum intensity level a given pixel if it is brighter than the mean intensity level of a 31×31 pixel window about it, and suppress it otherwise (`cvAdaptiveThreshold`).

2.1.2 Erode

To obtain a one-dimensional representation of the molecular backbone contours, we employ the erosion algorithm given in [12, 1], that applies a set of eight 3×3 pixel kernels as structuring elements to iteratively erode the binary regions of 8-connected pixels, halting when there is no change in the images of present and prior iterations. This process results in a set of 8-connected component edge pixels having unit thickness. (Figure 2b).

2.1.3 Select

The image is now a collection of 8-connected component edge pixels. We recursively traverse each component, labeling distinct branches, scoring them according to Euclidean distance from one pixel to the next: $\{N, S, E, W\} = 1, \{NW, NE, SW, SE\} = \sqrt{2}$. This results in a collection of weighted edge tree graphs. Finally, we identify the longest path through each edge tree graph, amounting to pruning branches from the trunk. The longest path represents the molecular backbone contour.

[2] http://www.wxwidgets.org/

[3] http://opencvlibrary.sourceforge.net/

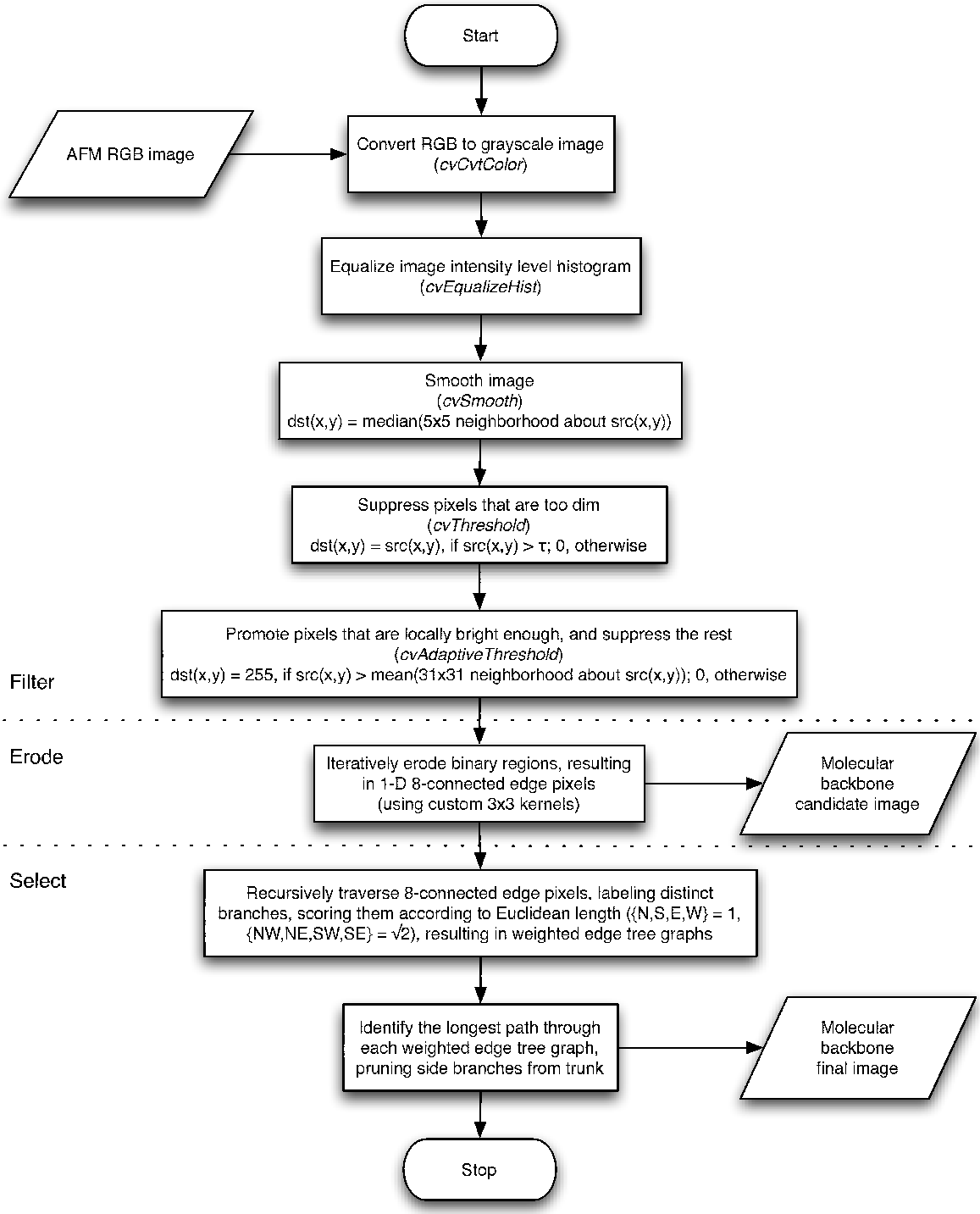


Figure 1: *AFM Explorer* image processing pipeline. An AFM image undergoes three phases of processing: (I) *filter* the noise using adaptive, local thresholding, yielding a set of two-dimensional binary image objects, (II) *erode* these into a set of one-dimensional binary image objects, and (III) *select* the longest path through each graph representation of the 8-connected component the final backbone image object.

Since we implemented a simplified algorithm from the one given in [5], we define it in Algorithms 1 and 2. Our algorithm is two consecutive breadth-first traversals across the 8-connected pixel graph. First, initiated from any extremity (deg = 1) pixel, e_1, a set of end-to-end pixel paths (with their associated computed lengths), $\mathcal{P}_{e_1}$, is constructed through a breadth-first traversal, branching at pixels having more than one unseen neighbor. Second, taking the terminal pixel, e_2, of the longest path from $\mathcal{P}_{e_1}$, another breadth-first traversal is initiated from e_2, constructing its respective set of end-to-end pixel paths, $\mathcal{P}_{e_2}$, in the same fashion. Upon completion, the longest path in $\mathcal{P}_{e_1} \cup \mathcal{P}_{e_2}$ is the longest path in the whole 8-connected pixel graph. (Figure 2c).

2.2 AFM Explorer length estimation pipeline

AFM Explorer uses the length estimation pipeline schematically presented in Figure 3. We outline the steps below.

2.2.1 Initial estimation using straight line segments

Let $\mathcal{B}$ be the set of all backbone pixel vectors in the image. After image processing, we compute the initial estimate of contour length for each $\vec{b} \in \mathcal{B}$ as the sum of its consecutive pixel-to-pixel straight line segments, given by

$$L_{LS}(\vec{b}) = \sum_{b_i, b_j \in \vec{b}, j=i+1} \left\{ \begin{array}{cc} \sqrt{2} & \text{for} \quad |b_{i_x} - b_{j_x}| = 1 \wedge |b_{i_y} - b_{j_y}| = 1 \\ 1 & \text{otherwise} \end{array} \right\}. \tag{1}$$

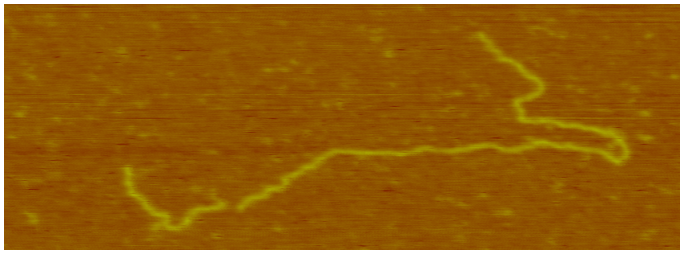
We then admit a subset $\mathcal{B}' \subset \mathcal{B}$ of backbone pixel vectors that meet two criteria, defined by

$$\mathcal{B}' = \{\vec{b}' \mid L_{LS}(\vec{b}') \in [min, max] \wedge \frac{|\vec{b}'|}{|c|} > \frac{1}{2}\}, \tag{2}$$

where min and max are set to some mode-dependent values, described below, and c is the 8-connected component from which $\vec{b}'$ is extracted. The second criterion is a heuristic to restrict intersecting backbones. The intuition is that in terms of numbers of pixels, $\vec{b}'$ should be at least half the size of c. This is built upon the following reasoning. Assume the average length per pixel of $\vec{b}'$ is equal to that of c. Suppose some pixel path $\vec{q} \in c \backslash \vec{b}'$ intersects $\vec{b}'$. Clearly $\vec{q}$ cannot intersect $\vec{b}'$ at its extremities, otherwise $\vec{q}$ would extend $\vec{b}'$, and that would contradict our assumption that $\vec{b}'$ is the longest path through c. So $\vec{q}$ must intersect $\vec{b}'$ in its interior. Further, the closer that intersection is to the midpoint of $\vec{b}'$, the longer $\vec{p}$ can be without contradiction. Thus, $|\vec{p}| < \frac{|\vec{b}'|}{2}$. Now suppose $\mathcal{Q}$ is a family of pixel paths, such that $c \backslash \vec{b}' = \bigcup_{\vec{q} \in \mathcal{Q}} .$ $\mathcal{Q}$ may take many configurations, but under the constraint that $\frac{|\vec{b}'|}{|c|} > \frac{1}{2}$, there can be at most two pixel paths, $\vec{q_1}, \vec{q_2} \in \mathcal{Q}$, stemming from the midpoint of $\vec{b}'$ having combined size $|\vec{q_1} \cup \vec{q_2}| < |\vec{b}'|$. This seems to us to be a reasonable upper bound on the tolerance for total intersection size, since at this size, $\vec{q_1}$ and $\vec{q_2}$ may still be reasonably classified as branch artifacts instead of overlapping molecules.

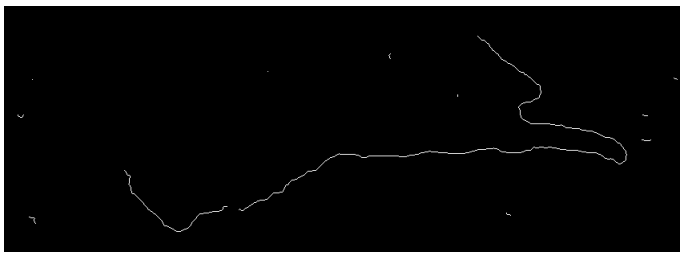
2.2.2 Secondary estimation using cubic spline fitting

Then, for each $\vec{b}' \in \mathcal{B}'$, we compute a sequence of cubic splines fitted to each consecutive 5-pixel subsequence, where the last pixel of a given subsequence is the first pixel of the next (i.e. all subsequences share one extremity pixel). A tailing subsequence, $\vec{b}'_t$, having $p < 5$ pixels is handled by fitting a cubic spline to the subsequence formed by prepending to $\vec{b}'_t$ the prior $5 - p$ pixels, then

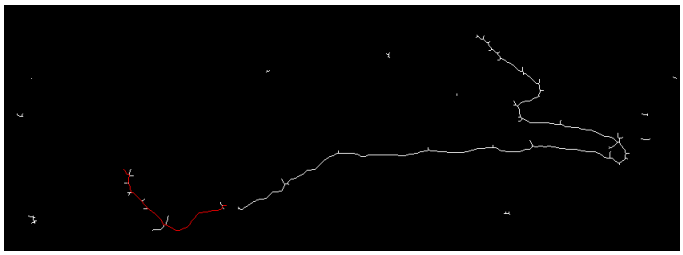


(a) The original 24-bit RGB AFM image.

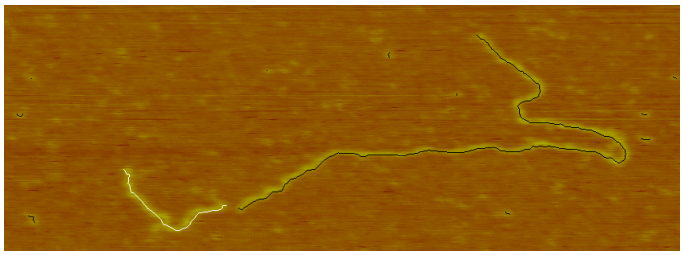
(b) The image after filtering and iterative erosion.



(c) The image after graph translation and backbone selection.



(d) A backbone (red) in the filtered and eroded image.



(e) A backbone (white) with other backbones (black) in the original image.

Figure 2: Results of the *AFM Explorer* image processing pipeline.

Algorithm 1 FIND-BACKBONES : $\mathcal{T} \rightarrow \mathcal{B}$

Let $\mathcal{T}$ be a set of 8-connected pixels forming an n-ary tree (i.e. an image object resulting from an iterative erosion algorithm).

Let $deg : t \in \mathcal{T} \rightarrow \mathbb{N} \cap [1,8] \equiv \sum_{dx \in \{-1,0,1\}, dy \in \{-1,0,1\}, dx \neq 0 \vee dy \neq 0} \left\{ \begin{array}{ccc} 1 & \text{for} & (t_x + dx, t_y + dy) \in \mathcal{T} \\ 0 & \text{otherwise} & \end{array} \right\}$ be the degree function that sums the number of pixels in $\mathcal{T}$ that are 8-connected with a given pixel in $\mathcal{T}$.

Let $\mathcal{E} = \{e \mid e \in \mathcal{T}, deg(e) = 1\}$ be the set of extremity pixels in $\mathcal{T}$.

Let $adj : t_a, t_b \in \mathcal{T} \rightarrow \{true, false\} \equiv |t_{a_x} - t_{b_x}| < 2 \wedge |t_{a_y} - t_{b_y}| < 2$ be the adjacency function that logically determines if two pixels in $\mathcal{T}$ are 8-connected with each other.

Let $\mathcal{P} = \{p \mid p = \langle p_1, p_2, ..., p_n \rangle, p_i \in \mathcal{T}, 1 \leq i \leq n, p_j \in \mathcal{E}, j \in \{1, n\}, p_k \notin \mathcal{E}, 1 < k < n, adj(p_a, p_b), 1 \leq a < n, b = a + 1\}$ be the set of end-to-end simple paths through $\mathcal{T}$.

Let $len : p \in \mathcal{P} \rightarrow \mathbb{R} \equiv \sum_{p_a, p_b \in p, b = a+1} \left\{ \begin{array}{ccc} \sqrt{2} & \text{for} & |p_{a_x} - p_{b_x}| = 1 \wedge |p_{a_y} - p_{b_y}| = 1 \\ 1 & \text{otherwise} & \end{array} \right\}$ be the length function that sums the straight line pixel transition distances in an end-to-end simple path through $\mathcal{T}$.

Let $\mathcal{B} = \{b \mid b \in \mathcal{P}, len(b) = \max_{p \in \mathcal{P}} len(p)\}$ be the set of longest end-to-end simple paths through $\mathcal{T}$ (i.e. a backbone contours).

$pid \leftarrow 0$
$\mathcal{S} \leftarrow \emptyset$
$p_{init}.len \leftarrow 0$
$p_{init}.pix \leftarrow \emptyset$
$\mathcal{P}_{e_1} \leftarrow \{p_{init}\}$
`Find-Paths-From`$(e_1 \in \mathcal{E}, \emptyset, \mathcal{T}, \vec{\mathcal{P}_{e_1}}, \vec{\mathcal{S}}, \vec{pid}, pid)$
$pid \leftarrow pid + 1$
$\mathcal{B}_{e_1} \leftarrow \{b \mid b \in \mathcal{P}_{e_1}, len(b) = \max_{p \in \mathcal{P}_{e_1}} len(p)\}$
$e_2 \leftarrow b.pix[last]$, where $b \in \mathcal{B}_{e_1}$
$\mathcal{S} \leftarrow \emptyset$
$p_{init}.len \leftarrow 0$
$p_{init}.pix \leftarrow \emptyset$
$\mathcal{P}_{e_2} \leftarrow \{p_{init}\}$
`Find-Paths-From`$(e_2, \emptyset, \mathcal{T}, \vec{\mathcal{P}_{e_2}}, \vec{\mathcal{S}}, \vec{pid}, pid)$
$\mathcal{B}_{e_2} \leftarrow \{b \mid b \in \mathcal{P}_{e_2}, len(b) = \max_{p \in \mathcal{P}_{e_2}} len(p)\}$
$\mathcal{B} \leftarrow \{b \mid b \in \mathcal{B}_{e_1} \cup \mathcal{B}_{e_2}, len(b) = \max_{b' \in \mathcal{B}_{e_1} \cup \mathcal{B}_{e_2}} len(b')\}$
return $\mathcal{B}$

Algorithm 2 FIND-PATHS-FROM : $pix \times dir \times \mathcal{T} \times \vec{\mathcal{P}} \times \vec{\mathcal{S}} \times \vec{pid} \times root_pid \to \emptyset$

Let $dir : p_a, p_b \in p \in \mathcal{P} \to \{N, S, E, W, NW, NE, SW, SE\} \equiv$
$$\left\{\begin{array}{rll} N & \text{for} & dx = 0 \wedge dy = 1 \\ S & \text{for} & dx = 0 \wedge dy = -1 \\ E & \text{for} & dx = 1 \wedge dy = 0 \\ W & \text{for} & dx = -1 \wedge dy = 0 \\ NW & \text{for} & dx = -1 \wedge dy = 1 \\ NE & \text{for} & dx = 1 \wedge dy = 1 \\ SW & \text{for} & dx = -1 \wedge dy = -1 \\ SE & \text{for} & dx = 1 \wedge dy = -1 \end{array}\right\}$$
be the direction function that gives the orientation of p_b with respect to p_a.

1: **if** $pix \in \vec{\mathcal{S}}$ **then**
2: **return**
3: $\vec{\mathcal{S}} \leftarrow \vec{\mathcal{S}} \cup \{pix\}$
4: **if** $dir \in \{N, S, E, W\}$ **then**
5: $\vec{\mathcal{P}}[root_pid].len \leftarrow \vec{\mathcal{P}}[root_pid].len + 1$
6: **else if** $dir \in \{NE, NW, SE, SW\}$ **then**
7: $\vec{\mathcal{P}}[root_pid].len \leftarrow \vec{\mathcal{P}}[root_pid].len + \sqrt{2}$
8: $\vec{\mathcal{P}}[root_pid].pix \leftarrow \vec{\mathcal{P}}[root_pid].pix \cup \{pix\}$
9: $\mathcal{N} \leftarrow \emptyset$
10: **for** $dx \in \{-1, 0, 1\}$ **do**
11: **for** $dy \in \{-1, 0, 1\}$ **do**
12: **if** $dx = 0 \wedge dy = 0$ **then**
13: continue
14: $n_pix \leftarrow (pix_x + dx, pix_y + dy)$
15: **if** $n_pix \in \vec{\mathcal{S}}$ **then**
16: continue
17: $n.pix \leftarrow n_pix$
18: $n.dir \leftarrow dir(dx, dy)$
19: $\mathcal{N} \leftarrow \mathcal{N} \cup \{n\}$
20: **for** $n \in \mathcal{N}$ **do**
21: **if** $|\mathcal{N}| > 1$ **then**
22: $\vec{pid} \leftarrow \vec{pid} + 1$
23: $\vec{\mathcal{P}}[last + 1] \leftarrow \vec{\mathcal{P}}[root_pid]$
24: `Find-Paths-From`$(n.pix, n.dir, \mathcal{T}, \vec{\mathcal{P}}, \vec{\mathcal{S}}, \vec{pid}, pid)$
25: **if** $|\mathcal{N}| > 1$ **then**
26: $\vec{\mathcal{P}}[root_pid] \leftarrow \emptyset$
27: **return**

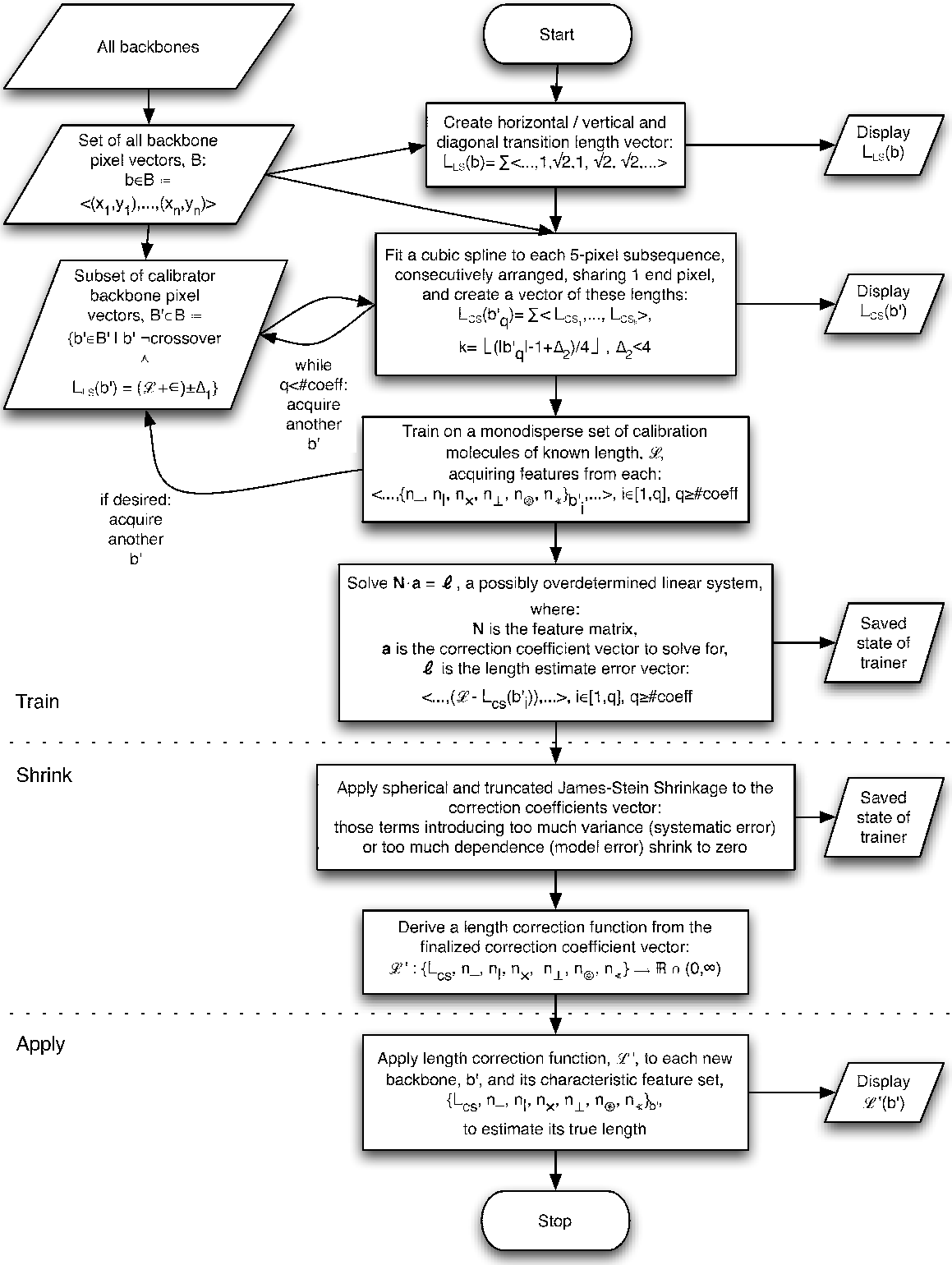


Figure 3: *AFM Explorer* length estimation pipeline. Length estimation undergoes three phases of processing: (I) *train* the coefficients of a simple linear regression model on a flexibly sized set of calibrating molecule backbones, each with its cubic spline length and set of features, (II) *shrink* the model coefficients according to how much variance they introduce, collectively (entailing spherical shrinkage) and individually (entailing truncated shrinkage), and (III) *apply* the trained and tuned model to correct the cubic spline length of novel backbones according to their individual features.

counting the spline's length from its closest approach to the first and last pixels in $\vec{b}'_t$. The resulting summed length of the cubic splines gives the second estimate of contour length, given by

$$L_{CS} = \sum_{i=1}^{k} \mathcal{CSL}_i, \tag{3}$$

where k is the number of cubic splines that fit the $|\vec{b}'|$ pixels, given by

$$k = \lfloor \frac{|\vec{b}'| - 1 + \delta}{4} \rfloor, \delta < 4, \tag{4}$$

and

$$\mathcal{CSL}_i = t_{i_\beta} - t_{i_\alpha}, \tag{5}$$

where α and β are the first and last pixels in the i^{th} 5-pixel subsequence being fit with a cubic spline, and t_{i_α} and t_{i_β} are the respective values of the length parameter t that satisfy, respectively

$$\frac{d}{dt}\left[\sqrt{(\alpha_x - x(t))^2 + (\alpha_y - y(t))^2}\right]\Bigg|_{t=t_{i_\alpha}} = 0 \tag{6}$$

and

$$\frac{d}{dt}\left[\sqrt{(\beta_x - x(t))^2 + (\beta_y - y(t))^2}\right]\Bigg|_{t=t_{i_\beta}} = 0, \tag{7}$$

(i.e. the values of t where the cubic spline makes its closest approach to the first and last pixels), where

$$x(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0 \tag{8}$$

and

$$y(t) = b_3 t^3 + b_2 t^2 + b_1 t + b_0 \tag{9}$$

form the parametric equations of the cubic spline, and are solutions to the respective overdetermined systems

$$\begin{bmatrix} t_1^3 & t_1^2 & t_1 & 1 \\ t_2^3 & t_2^2 & t_2 & 1 \\ t_3^3 & t_3^2 & t_3 & 1 \\ t_4^3 & t_4^2 & t_4 & 1 \\ t_5^3 & t_5^2 & t_5 & 1 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \Longleftrightarrow T\vec{a} = \vec{x} \tag{10}$$

and

$$\begin{bmatrix} t_1^3 & t_1^2 & t_1 & 1 \\ t_2^3 & t_2^2 & t_2 & 1 \\ t_3^3 & t_3^2 & t_3 & 1 \\ t_4^3 & t_4^2 & t_4 & 1 \\ t_5^3 & t_5^2 & t_5 & 1 \end{bmatrix} \begin{bmatrix} b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} \Longleftrightarrow T\vec{b} = \vec{y}, \tag{11}$$

where

$$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{bmatrix} = \begin{bmatrix} L_{LS}([(x_1, y_1)]) \\ L_{LS}([(x_1, y_1) \quad (x_2, y_2)]) \\ L_{LS}([(x_1, y_1) \quad (x_2, y_2) \quad (x_3, y_3)]) \\ L_{LS}([(x_1, y_1) \quad (x_2, y_2) \quad (x_3, y_3) \quad (x_4, y_4)]) \\ L_{LS}([(x_1, y_1) \quad (x_2, y_2) \quad (x_3, y_3) \quad (x_4, y_4) \quad (x_5, y_5)]) \end{bmatrix}, \tag{12}$$

and thus $\vec{a}$ and $\vec{b}$ can respectively be evaluated analytically by

$$\vec{a} = (T^T T)^{-1} T^T \vec{x} \tag{13}$$

and

$$\vec{b} = (T^T T)^{-1} T^T \vec{y}, \tag{14}$$

using the Tikhonov regularization of $(T^T T + \lambda I)^{-1}$, where $\lambda = 1 \times 10^{-5}$, in the case where no inverse exists for $(T^T T)$, for an arbitrary 5-pixel subsequence. To visualize how the spline fitting looks, in simulated and real situations, refer to Figure 4.

The pipeline has three phases (or rather, operates in three distinct modes): train, shrink, and apply.

2.2.3 Train

When the application runs in train mode, each admissible backbone pixel vector, $\vec{b'} \in \mathcal{B}'$, its cubic spline contour length estimate, $L_{CS}(\vec{b'})$, and its computed feature values (described below) form the data of a possibly overdetermined linear system. We assume the images used to train represent a monodisperse set of molecules having known theoretical length $\mathcal{L}$. Accordingly, the values of *min* and *max* should reflect reasonable expectations for a spread of $L_{LS}(\vec{b'})$ values observed for these molecules. For example, in one of our experiments, we trained on images of monodisperse cDNAs having theoretical length 75 nm. Since we have empirically observed at least +10 nm translation of the mean due to systematic errors, we chose a mean of 85 nm and a spread of ± 15 nm — thus, we set *min* to 70 nm and *max* to 100 nm.

Claim 1. *Since each fit cubic spline locally minimizes the sum of squares error in its 5-pixel window, then our conjoined cubic spline fitting across the whole backbone contour is a best linear unbiased estimator (BLUE), and that any further correction to the estimate is systematic error,*

$$\mathcal{L} - L_{CS}(\vec{b'}) = \varepsilon_{\vec{b'}}, \tag{15}$$

that can be derived from backbone-length-dependent feature values we learn from the data.

Claim 2. *Given an increasingly large training sample and an increasingly rich feature set, our estimator will approach the optimal estimator.*

The features. We considered 8 features for our modeling of the systematic error. Given $\vec{b'} \in \mathcal{B}'$:

Definition 1. *The* number of horizontal pixel pairs, $n_{horz} : \vec{b'} \to \mathbb{N} \equiv$
$$\sum_{b_i, b_j \in \vec{b'}, j=i+1} \left\{ \begin{array}{ll} 1 & \textit{for} \quad |b_{i_x} - b_{j_x}| = 1 \wedge |b_{i_y} - b_{j_y}| = 0 \\ 0 & \textit{otherwise} \end{array} \right\}.$$

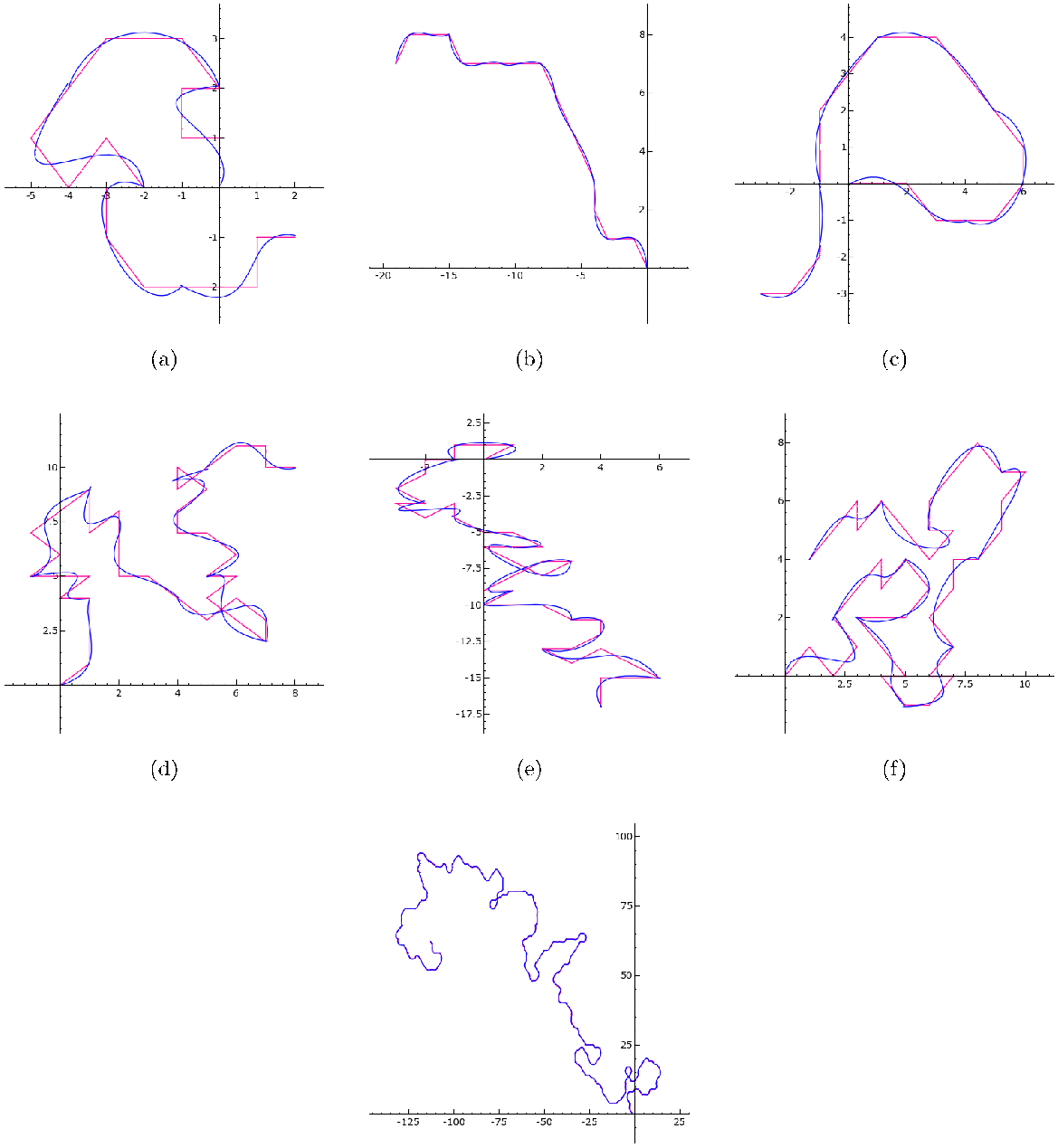
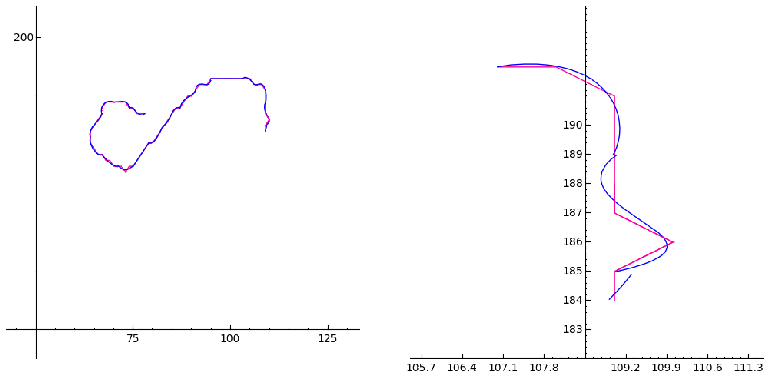


Figure 4: Simulated molecules as n-pixel, nonoverlapping random walk sequences (red) fitted by $\lfloor \frac{n-1+\delta}{4} \rfloor$, $\delta < 4$, consecutive cubic splines (blue). Each cubic spline fits to a 5-pixel subsequence; consecutive cubic splines share end points. (4a, 4b, 4c): 21 pixels fitted by 5 cubic splines. (4d, 4e, 4f): 41 pixels fitted by 10 cubic splines. (4g): 101 pixels fitted by 100 cubic splines [$L_{LS} = 479.94$ nm, $L_{CS} = 480.09$ nm].



(a) (b)

Figure 5: (5a): Real cDNA molecule as a 70-pixel sequence (red) fitted by 18 consecutive cubic splines (blue). (5b): A detail view of the last 3 spline fits. Each cubic spline fits to a 5-pixel subsequence; consecutive cubic splines share end points. Note that in 5b, since the last subsequence to fit has only 2 pixels, the prior $5 - 2 = 3$ pixels are prepended to it to fit the spline, and then only the relevant section of that spline is used. [$\mathcal{L} = 75$ nm, $L_{LS} = 79.92$ nm, $L_{CS} = 79.78$ nm.]

DEFINITION 2. *The* number of vertical pixel pairs, $n_{vert} : \vec{b'} \rightarrow \mathbb{N} \equiv$
$$\sum_{b_i, b_j \in \vec{b'}, j=i+1} \left\{ \begin{array}{lll} 1 & for & |b_{i_x} - b_{j_x}| = 0 \wedge |b_{i_y} - b_{j_y}| = 1 \\ 0 & otherwise & \end{array} \right\}.$$

DEFINITION 3. *The* number of diagonal pixel pairs, $n_{diag} : \vec{b'} \rightarrow \mathbb{N} \equiv$
$$\sum_{b_i, b_j \in \vec{b'}, j=i+1} \left\{ \begin{array}{lll} 1 & for & |b_{i_x} - b_{j_x}| = 1 \wedge |b_{i_y} - b_{j_y}| = 1 \\ 0 & otherwise & \end{array} \right\}.$$

DEFINITION 4. *The* number of perpendicular pixel triplets, $n_{perp} : \vec{b'} \rightarrow \mathbb{N} \equiv$
$$\sum_{b_i, b_j, b_k \in \vec{b'}, j=i+1, k=j+1} \left\{ \begin{array}{lll} & & \left\{ \begin{array}{r} (|b_{i_x} - b_{j_x}| = 1 \wedge |b_{i_y} - b_{j_y}| = 0) \\ \wedge \\ (|b_{j_x} - b_{k_x}| = 0 \wedge |b_{j_y} - b_{k_y}| = 1) \end{array} \right\} \\ & & \vee \\ 1 & for & \left\{ \begin{array}{r} (|b_{i_x} - b_{j_x}| = 0 \wedge |b_{i_y} - b_{j_y}| = 1) \\ \wedge \\ (|b_{j_x} - b_{k_x}| = 1 \wedge |b_{j_y} - b_{k_y}| = 0) \end{array} \right\} \\ & & \vee \\ & & \left\{ \begin{array}{r} (|b_{i_x} - b_{j_x}| = 1 \wedge |b_{i_y} - b_{j_y}| = 1) \\ \wedge \\ (|b_{j_x} - b_{k_x}| = 1 \wedge |b_{j_y} - b_{k_y}| = 1) \end{array} \right\} \\ 0 & otherwise & \end{array} \right\}.$$

DEFINITION 5. *The* mean backbone height, $n_{htav} : \vec{b'} \rightarrow \mathbb{R} \equiv \frac{1}{|\vec{b'}|} \sum_{b \in \vec{b'}} I_{gry}(b)$, *where* $I_{gry} : b \rightarrow \mathbb{N} \cap [0, 255]$ *gives the intensity value for pixel* b *in the grayscale image.*

DEFINITION 6. *The* standard deviation of backbone height, $n_{htsd} : \vec{b'} \rightarrow \mathbb{R} \equiv$

$\sqrt{\frac{1}{|\vec{b'}|}\sum_{b\in\vec{b'}}(I_{gry}(b)-n_{htav}(\vec{b'}))^2}$.

DEFINITION 7. *The* mean backbone thickness, $n_{tkav}:\vec{b'}\rightarrow\mathbb{R}\equiv\frac{1}{|\vec{b'}|}\sum_{b\in\vec{b'}}M(b)$, *where*

$$M:b\rightarrow\mathbb{R}\equiv\min\left\{\begin{array}{l}\sum\limits_{\{p\ |\ p_x=b_x-1,b_x-2,...\wedge p_y=b_y\wedge I_{bin}(p)=255\}}1+\\ \sum\limits_{\{p\ |\ p_x=b_x+1,b_x+2,...\wedge p_y=b_y\wedge I_{bin}(p)=255\}}1,\\ \\ \sum\limits_{\{p\ |\ p_x=b_x\wedge p_y=b_y-1,b_y-2,...\wedge I_{bin}(p)=255\}}1+\\ \sum\limits_{\{p\ |\ p_x=b_x\wedge p_y=b_y+1,b_y+2,...\wedge I_{bin}(p)=255\}}1,\\ \\ \sum\limits_{\{p\ |\ p_x=b_x-1,b_x-2,...\wedge p_y=b_y-1,b_y-2,...\wedge I_{bin}(p)=255\}}\sqrt{2}+\\ \sum\limits_{\{p\ |\ p_x=b_x+1,b_x+2,...\wedge p_y=b_y+1,b_y+2,...\wedge I_{bin}(p)=255\}}\sqrt{2},\\ \\ \sum\limits_{\{p\ |\ p_x=b_x-1,b_x-2,...\wedge p_y=b_y+1,b_y+2,...\wedge I_{bin}(p)=255\}}\sqrt{2}+\\ \sum\limits_{\{p\ |\ p_x=b_x+1,b_x+2,...\wedge p_y=b_y-1,b_y-2,...\wedge I_{bin}(p)=255\}}\sqrt{2}\end{array}\right\}$$

gives the minimum of four pairs of linear distances that radiate in opposite directions from the origin at pixel b until they reach the edge of the binary object, thereby covering the 8 cardinal directions connecting the pixels in the binary object, where $I_{bin}:p\rightarrow\{0,255\}$ *gives the intensity value for pixel p in the binary image.*

DEFINITION 8. *The* standard deviation of backbone thickness, $n_{tksd}:\vec{b'}\rightarrow\mathbb{R}\equiv\sqrt{\frac{1}{|\vec{b'}|}\sum_{b\in\vec{b'}}(M(b)-n_{tkav}(\vec{b'}))^2}$.

The model. We train a linear regression model on $q\geq 5$ calibrating molecule backbones, $\vec{b'}\in\mathcal{B'}$, having known theoretical length $\mathcal{L}$, using values from 5 of these 8 features (for reasons discussed below): $\{n_{horz},n_{vert},n_{diag},n_{perp},n_{tkav}\}$, giving

$$\begin{bmatrix}n_{horz}(\vec{b'_1}) & n_{vert}(\vec{b'_1}) & n_{diag}(\vec{b'_1}) & n_{perp}(\vec{b'_1}) & n_{tkav}(\vec{b'_1})\\ n_{horz}(\vec{b'_2}) & n_{vert}(\vec{b'_2}) & n_{diag}(\vec{b'_2}) & n_{perp}(\vec{b'_2}) & n_{tkav}(\vec{b'_2})\\ n_{horz}(\vec{b'_3}) & n_{vert}(\vec{b'_3}) & n_{diag}(\vec{b'_3}) & n_{perp}(\vec{b'_3}) & n_{tkav}(\vec{b'_3})\\ n_{horz}(\vec{b'_4}) & n_{vert}(\vec{b'_4}) & n_{diag}(\vec{b'_4}) & n_{perp}(\vec{b'_4}) & n_{tkav}(\vec{b'_4})\\ n_{horz}(\vec{b'_5}) & n_{vert}(\vec{b'_5}) & n_{diag}(\vec{b'_5}) & n_{perp}(\vec{b'_5}) & n_{tkav}(\vec{b'_5})\\ ... & ... & ... & ... & ...\\ n_{horz}(\vec{b'_q}) & n_{vert}(\vec{b'_q}) & n_{diag}(\vec{b'_q}) & n_{perp}(\vec{b'_q}) & n_{tkav}(\vec{b'_q})\end{bmatrix}\begin{bmatrix}a_1\\ a_2\\ a_3\\ a_4\\ a_5\end{bmatrix}=\begin{bmatrix}l_1\\ l_2\\ l_3\\ l_4\\ l_5\\ ...\\ l_q\end{bmatrix}\iff N\vec{a}=\vec{l},\qquad(16)$$

where N is the $q\times 5$ feature matrix, $\vec{a}$ is the correction coefficient 5-vector to solve for, and $\vec{l}$ is the length estimate error q-vector $[...,(\mathcal{L}-L_{CS}(\vec{b'_i})),...]$, where $i=1,...,q$. The model has the analytic solution

$$\vec{a}=(N^TN)^{-1}N^T\vec{l},\qquad(17)$$

using the Tikhonov regularization of $(N^T N + \lambda I)^{-1}$, where $\lambda = 1 \times 10^{-5}$, in the case where no inverse exists for $(N^T N)$.

Once $q \geq 5$, the application will solve for $\vec{a}$ and save its values to disk so that the trained model can be applied to test data before it undergoes James-Stein shrinkage. This facilitates comparisons against results associated with shrunken coefficient values, to interpret the effectiveness of the next phase, or mode, of processing.

2.2.4 Shrink

In our modeling of estimation error above, one or more features in training may introduce too much variance (systematic error) or dependence (model error). We would like our model to have an extensible and adaptive structure, where any number of features may be used, and proceed with confidence, knowing that noisy or dependent features will have a contribution to the estimate that shrinks to zero. In shrink mode, the application simply applies one of the following patterns of shrinkage to the correction coefficients, $\vec{a}$, without applying the resulting backbone contour length estimator to test data — the task of apply mode, described below.

2.2.5 James-Stein shrinkage

In 1961, James and Stein published their seminal paper [20] describing a method to improve estimating a multivariate normal mean, $\vec{\mu} = [\mu_1, ..., \mu_k]$, under expected sum of squares error loss, provided the degree of freedom $k \geq 3$, and the μ_i are close to the point to which the improved estimator shrinks.

Spherical shrinkage. Let $\vec{a} = [a_i, ..., a_k]$ have a k-variate normal distribution with mean vector $\vec{\mu}$ and covariance matrix $\sigma^2 I$, which we measure empirically in train mode. We would like to estimate $\vec{\mu}$ using an estimator

$$\delta(\vec{a}) = [\delta_1(\vec{a}), ..., \delta_k(\vec{a})] \tag{18}$$

under the sum of squares error loss

$$L(\vec{\mu}, \delta) = \sum_{i=1}^{k} (\mu_i - \delta_i)^2 \tag{19}$$

In terms of expected loss,

$$R(\vec{\mu}, \delta) = E_\mu[L(\vec{\mu}, \delta(\vec{a}))], \tag{20}$$

the result of [20] shows that when $k \geq 3$, an improved estimator is obtained by a symmetric (or spherical) shrinkage in $\vec{a}$ given by

$$\delta(\vec{a}) = \left[1 - \frac{\kappa(q-k)s^2}{\sum_{i=1}^{q} (N\vec{a})_i^2} \right] \vec{a}, \tag{21}$$

where

$$\kappa = \frac{(k-2)}{(q-k+2)}, \tag{22}$$

and s^2 is the empirical estimate of variance, σ^2, given by

$$s^2 = \frac{1}{(q-k)} \sum_{i=1}^{q} (\mathcal{L} - L_{CS_i} - (N\vec{a})_i)^2. \tag{23}$$

Shortly afterward, Stanley Sclove published a modified form of this estimator that treats negative coefficient values as zero, given by

$$\delta(\vec{a}) = \left[1 - \frac{\kappa(q-k)s^2}{\sum_{i=1}^{q} (N\vec{a})_i^2} \right]^+ \vec{a}, \tag{24}$$

where $[x]^+ \equiv max\{0, x\}$.

Truncated shrinkage. When extreme μ_i are likely, then spherical shrinkage may give little improvement. This may occur, for instance, when the μ_i arise from a prior distribution with a long tail. A property of (24) is that its performance is guaranteed only in a small subspace of parameter space [8], requiring that one select an estimator designed with some notion of where $\vec{\mu}$ is likely to be, such that the estimator shrinks toward it. An extreme μ_i will likely be outside of any small selected subspace, implying a large denominator and so little, if any, shrinkage in $\vec{a}$, thereby giving no improvement. To address this problem, Stein [35] proposed a coordinate-based (or truncated) shrinkage method, given by

$$\delta_i^{(f)}(\vec{a}) = \left[1 - \frac{(f-2)s^2 min\{1, \frac{z_{(f)}}{|a_i|}\}}{\sum_{j=1}^{q} (N\vec{m})_j^2} \right]^+ a_i, \tag{25}$$

where f is a "large fraction" [8] of k, $z_i = |a_i|, i = 1, ..., k$, $z_{(1)} < z_{(2)} < ... < z_{(f)} < ... < z_{(k)}$ forms a strictly increasing ordering on $z_1, ..., z_k$, s^2 is the empirical estimate of variance, σ^2, given by

$$s^2 = \frac{1}{(q-k)} \sum_{i=1}^{q} (\mathcal{L} - L_{CS_i} - (N\vec{a})_i)^2, \tag{26}$$

and $\vec{m}_i = min\{a_i, z_{(f)}\}, i = 1, ..., k$. The result in [35] shows this estimator is minimax if $f \geq 3$. Observe that the denominator is small even when $(k - f)$ of the μ_i are extreme.

Once the application has trained the model, and saved the correction coefficients, $\vec{a}$, to disk, the application may then separately apply spherical or truncated James-Stein shrinkage, and save the appropriate shrunken correction coefficients, $\delta(\vec{a})$ or $\delta_i^{(f)}(\vec{a}), i = 1, ..., k$, to disk for later comparison and use against test data — the task of apply mode, described below.

2.2.6 Apply

When the application is in apply mode, the model correction coefficients are locked — either they are unadjusted from training, or have been adjusted by the spherical or truncated shrinkage method — and are loaded from disk. In this mode, *min* and *max* are set to admit all molecules having a reasonable backbone contour length as first estimated by L_{LS}. So $min = 40$ nm and $max = 1000$ nm makes for a good range in most test situations. Then each $\vec{b}' \in \mathcal{B}'$ obtains its final estimate, $\mathcal{L}'$, from the correction function

$$C : \mathcal{B}' \to \mathbb{R}$$
$$: \vec{b}' \mapsto \tag{27}$$
$$a_1 n_{horz}(\vec{b}') + a_2 n_{vert}(\vec{b}') + a_3 n_{diag}(\vec{b}') + a_4 n_{perp}(\vec{b}') + a_5 n_{tkav}(\vec{b}').$$

and is given by

$$\mathcal{L}'(\vec{b}') = L_{CS}(\vec{b}') + C(\vec{b}'). \tag{28}$$

We presently discuss the experimental results of our model's performance ($\mathcal{L} - \mathcal{L}'$), and related factors, on a large set of training and test images.

3 Results

An early version of *AFM Explorer* reported L_{LS} for all existing fragments in the image. Comparing these automatically computed values with the length estimates of hand-drawn backbones (Figure 6) gave us reason to believe that while an image processing pipeline can bring us close to the apparent length of DNAs and RNAs, more would be required. Namely, bridging the gap between apparent and true length would first require using a better length estimator (L_{CS}), and then from that, modeling the systematic error intrinsic to the problem.

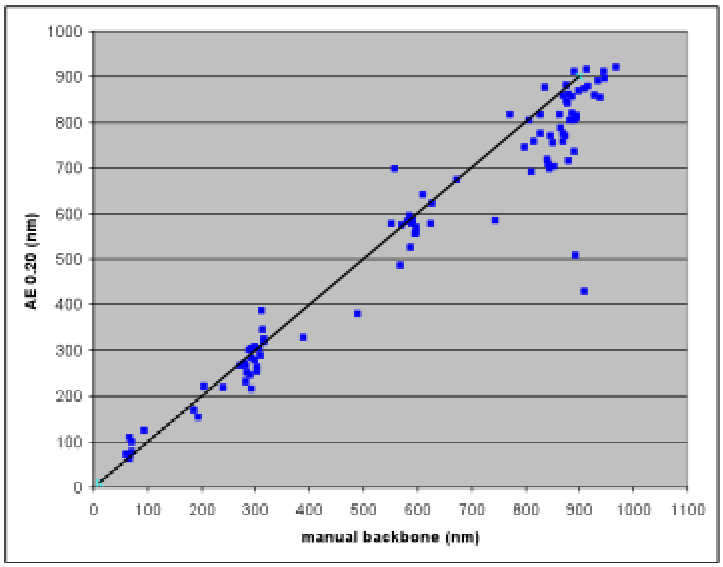


Figure 6: Early comparative results. Monodisperse pUC19 plasmids were linearized with EcoRi and digested with RsaI restriction enzymes. Fifty AFM images were taken of the resulting fragments, from which about 100 fragments were selected and tagged. The lengths given by *AFM Explorer* (version 0.20, producing piecewise line segment lengths, L_{LS}) were compared against those of hand-drawn backbones using *NIH Image*. Note that as length increased, automatically computed L_{LS} (labeling the y-axis as "AE 0.20") progressively underestimated fragment backbone length. Note too the proximity of clustering (and theoretically given cleavage points induced by RsaI) around 90 (75), 275 (223), and 580 (584) nm; the clustering around 800 nm suggested failed digestion (an intrinsic experimental error).

Here we explain a number of results, beginning with three regression models we attempted to use, but failed because of overfitting during the training phase; then we present the current model we used to obtain close estimates, which did not suffer from overfitting.

3.1 Inadmissible models

The following three models overfitted the training data.

3.1.1 Quadratic 8-feature

We trained a quadratic regression model on $q \geq 36$ calibrating molecule backbones, $\vec{b}' \in \mathcal{B}'$, having known theoretical length $\mathcal{L}$, using values from all 8 features defined above: $\{n_{horz}, n_{vert}, n_{diag}, n_{perp}, n_{htav}, n_{htsd}, n_{tkav}, n_{tksd}\}$, giving

$$\begin{bmatrix} n_{horz}(\vec{b}'_1)^2 & ... & n_{tksd}(\vec{b}'_1)^2 & n_{horz}(\vec{b}'_1)n_{vert}(\vec{b}'_1) & ... & n_{tksav}(\vec{b}'_1)n_{tksd}(\vec{b}'_1) \\ ... & ... & ... & ... & ... & ... \\ n_{horz}(\vec{b}'_{36})^2 & ... & n_{tksd}(\vec{b}'_{36})^2 & n_{horz}(\vec{b}'_{36})n_{vert}(\vec{b}'_{36}) & ... & n_{tkav}(\vec{b}'_{36})n_{tksd}(\vec{b}'_{36}) \\ ... & ... & ... & ... & ... & ... \\ n_{horz}(\vec{b}'_q)^2 & ... & n_{tksd}(\vec{b}'_q)^2 & n_{horz}(\vec{b}'_q)n_{vert}(\vec{b}'_q) & ... & n_{tkav}(\vec{b}'_q)n_{tksd}(\vec{b}'_q) \end{bmatrix} \begin{bmatrix} a_1 \\ ... \\ a_{36} \end{bmatrix} = \begin{bmatrix} l_1 \\ ... \\ l_{36} \\ ... \\ l_q \end{bmatrix}$$
$$\iff N\vec{a} = \vec{l}, \tag{29}$$

where N is the $q \times 36$ feature matrix, $\vec{a}$ is the correction coefficient 36-vector to solve for, and $\vec{l}$ is the length estimate error q-vector $[..., (\mathcal{L}^2 - L_{CS}(\vec{b}'_i)^2), ...]$, where $i = 1, ..., q$. The model has the analytic solution

$$\vec{a} = (N^T N)^{-1} N^T \vec{l}, \tag{30}$$

using the Tikhonov regularization of $(N^T N + \lambda I)^{-1}$, where $\lambda = 1 \times 10^{-5}$, in the case where no inverse exists for $(N^T N)$. Then each $\vec{b}' \in \mathcal{B}'$ obtains its final estimate, $\mathcal{L}'$, from the correction function

$$\begin{aligned} C : \mathcal{B}' &\to \mathbb{R} \\ : \vec{b}' &\mapsto \\ & a_1 n_{horz}(\vec{b}')^2 + a_2 n_{vert}(\vec{b}')^2 + a_3 n_{diag}(\vec{b}')^2 + a_4 n_{perp}(\vec{b}')^2 + \\ & a_5 n_{htav}(\vec{b}')^2 + a_6 n_{htsd}(\vec{b}')^2 + a_7 n_{tkav}(\vec{b}')^2 + a_8 n_{tksd}(\vec{b}')^2 + \\ & a_9 n_{horz}(\vec{b}')n_{vert}(\vec{b}') + a_{10} n_{horz}(\vec{b}')n_{diag}(\vec{b}') + a_{11} n_{horz}(\vec{b}')n_{perp}(\vec{b}') + a_{12} n_{horz}(\vec{b}')n_{htav}(\vec{b}') + \\ & a_{13} n_{horz}(\vec{b}')n_{htsd}(\vec{b}') + a_{14} n_{horz}(\vec{b}')n_{tkav}(\vec{b}') + a_{15} n_{horz}(\vec{b}')n_{tksd}(\vec{b}') + a_{16} n_{vert}(\vec{b}')n_{diag}(\vec{b}') + \\ & a_{17} n_{vert}(\vec{b}')n_{perp}(\vec{b}') + a_{18} n_{vert}(\vec{b}')n_{htav}(\vec{b}') + a_{19} n_{vert}(\vec{b}')n_{htsd}(\vec{b}') + a_{20} n_{vert}(\vec{b}')n_{tkav}(\vec{b}') + \\ & a_{21} n_{vert}(\vec{b}')n_{tksd}(\vec{b}') + a_{22} n_{diag}(\vec{b}')n_{perp}(\vec{b}') + a_{23} n_{diag}(\vec{b}')n_{htav}(\vec{b}') + a_{24} n_{diag}(\vec{b}')n_{htsd}(\vec{b}') + \\ & a_{25} n_{diag}(\vec{b}')n_{tkav}(\vec{b}') + a_{26} n_{diag}(\vec{b}')n_{tksd}(\vec{b}') + a_{27} n_{perp}(\vec{b}')n_{htav}(\vec{b}') + a_{28} n_{perp}(\vec{b}')n_{htsd}(\vec{b}') + \\ & a_{29} n_{perp}(\vec{b}')n_{tkav}(\vec{b}') + a_{30} n_{perp}(\vec{b}')n_{tksd}(\vec{b}') + a_{31} n_{htav}(\vec{b}')n_{htsd}(\vec{b}') + a_{32} n_{htav}(\vec{b}')n_{tkav}(\vec{b}') + \\ & a_{33} n_{htav}(\vec{b}')n_{tksd}(\vec{b}') + a_{34} n_{htsd}(\vec{b}')n_{tkav}(\vec{b}') + a_{35} n_{htsd}(\vec{b}')n_{tksd}(\vec{b}') + a_{36} n_{tkav}(\vec{b}')n_{tksd}(\vec{b}'), \end{aligned} \tag{31}$$

(i.e. the set of $\binom{8}{1} = 8$ quadratic feature terms and $\binom{8}{2} = 28$ linear pairwise feature interaction terms, totaling 36 model terms) and is given by

$$\mathcal{L}'(\vec{b}')^2 = L_{CS}(\vec{b}')^2 + C(\vec{b}'). \tag{32}$$

Table 1: Inadmissible (overfitted) feature and length estimation results for the Quadratic 8-feature model.

	Train Knowns $\mathcal{L} = 75$ nm 0.97 $\frac{nm}{pixel}$ $L_{LS} \in [70, 100]$ nm 5 images 275 cDNAs		Test Knowns $\mathcal{L} = 75$ nm 0.97 $\frac{nm}{pixel}$ $L_{LS} \in [70, 100]$ nm 14 images 782 cDNAs		Test Unknowns A $\mathcal{L} = 223$ nm 0.97 $\frac{nm}{pixel}$ $L_{LS} \in [70, 100]$ nm 44 images 22 cDNAs		Test Unknowns B $\mathcal{L} = 584$ nm 0.97 $\frac{nm}{pixel}$ $L_{LS} \in [70, 100]$ nm 101 images 139 cDNAs	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
n_{horz}	18.96	11.10	19.82	9.74	26.54	10.63	20.19	11.90
n_{vert}	21.04	12.27	20.56	11.21	14.13	10.38	19.06	11.53
n_{diag}	33.39	5.31	34.68	5.43	28.95	4.83	31.94	6.30
n_{perp}	1.42	1.30	1.54	1.24	2.13	1.76	2.19	1.80
n_{htav}	242.72	2.29	244.03	2.24	247.80	3.16	248.65	4.11
n_{htsd}	4.92	0.85	4.66	0.83	5.03	1.09	4.18	1.79
n_{tkav}	5.15	1.30	6.13	1.32	8.65	1.95	8.72	4.39
n_{tksd}	1.84	0.72	1.95	0.78	3.60	0.86	3.52	1.92
L_{LS}	85.19	6.81	87.34	7.22	79.71	6.99	82.45	8.30
L_{CS}	85.23	6.81	87.36	7.22	79.73	6.98	82.46	8.30
$\mathcal{L}'_{train}$	74.99	0.33	74.88	0.63	75.07	1.08	76.05	1.93
$\mathcal{L}'_{spherical}$	75.00	0.33	74.90	0.63	75.08	1.07	76.06	1.93
$\mathcal{L}'_{truncated}(f = 36)$	75.00	0.33	74.90	0.63	75.08	1.07	76.06	1.93
$\mathcal{L}'_{truncated}(f = 35)$	75.00	0.33	74.90	0.63	75.08	1.07	76.06	1.93
$\mathcal{L}'_{truncated}(f = 34)$	75.02	0.33	74.92	0.63	75.08	1.07	76.06	1.91
$\mathcal{L}'_{truncated}(f = 33)$	75.03	0.33	74.94	0.63	75.09	1.07	76.07	1.91
$\mathcal{L}'_{truncated}(f = 32)$	75.06	0.33	74.97	0.63	75.11	1.06	76.09	1.91
$\mathcal{L}'_{truncated}(f = 31)$	75.06	0.33	74.97	0.63	75.11	1.06	76.10	1.91
$\mathcal{L}'_{truncated}(f = 30)$	75.19	0.35	75.13	0.63	75.20	1.03	76.21	1.91
$\mathcal{L}'_{truncated}(f = 29)$	75.23	0.36	75.18	0.63	75.22	1.02	76.24	1.92
$\mathcal{L}'_{truncated}(f = 28)$	75.38	0.40	75.37	0.65	75.33	1.00	76.38	1.93
$\mathcal{L}'_{truncated}(f = 27)$	75.38	0.42	75.39	0.66	75.34	0.98	76.39	1.94
$\mathcal{L}'_{truncated}(f = 26)$	75.34	0.43	75.36	0.66	75.30	0.98	76.36	1.95
$\mathcal{L}'_{truncated}(f = 25)$	75.29	0.44	75.32	0.67	75.25	0.97	76.31	1.96
$\mathcal{L}'_{truncated}(f = 24)$	75.24	0.44	75.26	0.67	75.20	0.97	76.26	1.96
$\mathcal{L}'_{truncated}(f = 23)$	75.02	0.45	75.05	0.67	75.00	0.96	76.05	1.98
$\mathcal{L}'_{truncated}(f = 22)$	74.90	0.45	74.94	0.67	74.90	0.95	75.94	1.99
$\mathcal{L}'_{truncated}(f = 21)$	74.78	0.46	74.82	0.67	74.79	0.95	75.82	2.00
$\mathcal{L}'_{truncated}(f = 20)$	73.98	0.53	74.05	0.71	74.03	0.94	75.06	2.09

We trained this model on 275 cDNA molecules, in 5 images, having known theoretical length of 75 nm, using $min = 70$ and $max = 100$, assuming a mean of 85 nm, given our prior observation of a +10 nm mean translation due to tip dilation effects and thermal drift. We then tested the trained model on three datasets: (1) 782 cDNA molecules, in 14 images, having known theoretical length of 75 nm, using $min = 70$ and $max = 100$; (2) 22 cDNA molecules, in 44 images, having unknown theoretical length, using $min = 70$ and $max = 100$; and (3) 139 cDNA molecules, in 101 images, having unknown theoretical length, using $min = 70$ and $max = 100$. For each test we obtained mean and standard deviation values for all features, L_{LS}, L_{CS}, $\mathcal{L}'_{train}$, $\mathcal{L}'_{spherical}$, and $\mathcal{L}'_{truncated}$ for a range of f values. These results are summarized in Table 1 and presented graphically in Figure 7.

Our methodological flaw was in applying trained and trained-shrunken models to $min = 70$ and $max = 100$. As the data attest, in this range of cDNA sizes, the error rates look promisingly low. Had we opened up the range to, say, $min = 40$ and $max = 1000$, then we would have immediately recognized the problem of overfitting, as all mean $\mathcal{L}'$ values were 75 nm, with a standard deviation of 1 nm, irrespective of corresponding values of L_{LS} and L_{CS}. Now that we could properly identify when overfitting was occurring, we tried three more models. The first two failed in nearly the identical fashion, both computing mean $\mathcal{L}'_{train}$ values of 75 nm, with standard deviation of 1 nm, for all admitted molecules — we did not bother to compute $\mathcal{L}'_{spherical}$ or $\mathcal{L}'_{truncated}$. The third succeeded in computing all mean $\mathcal{L}'$ values without overfitting the training data, discussed below.

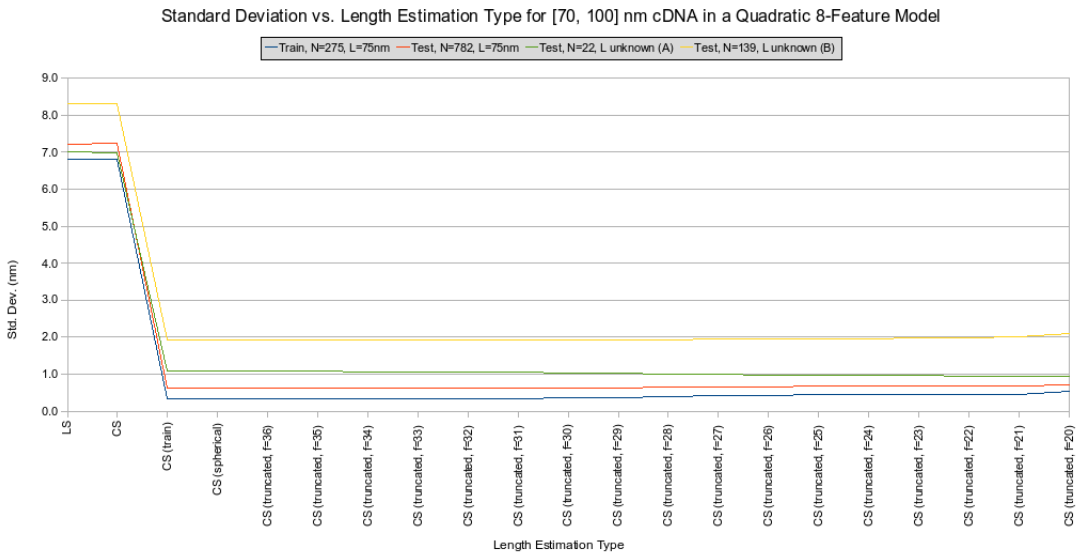


Figure 7: Inadmissible (overfitted) length estimation results for the Quadratic 8-feature model.

3.1.2 Linear 8-feature

We were first suspicious that the quadratic terms, and quadratic magnitudes of the linear pairwise terms, of the Quadratic 8-feature model were responsible for overfitting, so we created a linear 8-feature one to exclude them.

We trained a linear regression model on $q \geq 8$ calibrating molecule backbones, $\vec{b}' \in \mathcal{B}'$, having known theoretical length $\mathcal{L}$, using values from all 8 features defined above: $\{n_{horz}, n_{vert}, n_{diag}, n_{perp}, n_{htav}, n_{htsd}, n_{tkav}, n_{tksd}\}$, giving

$$\begin{bmatrix} n_{horz}(\vec{b}'_1) & n_{vert}(\vec{b}'_1) & n_{diag}(\vec{b}'_1) & n_{perp}(\vec{b}'_1) & n_{htav}(\vec{b}'_1) & n_{htsd}(\vec{b}'_1) & n_{tkav}(\vec{b}'_1) & n_{tksd}(\vec{b}'_1) \\ n_{horz}(\vec{b}'_2) & n_{vert}(\vec{b}'_2) & n_{diag}(\vec{b}'_2) & n_{perp}(\vec{b}'_2) & n_{htav}(\vec{b}'_2) & n_{htsd}(\vec{b}'_2) & n_{tkav}(\vec{b}'_2) & n_{tksd}(\vec{b}'_2) \\ n_{horz}(\vec{b}'_3) & n_{vert}(\vec{b}'_3) & n_{diag}(\vec{b}'_3) & n_{perp}(\vec{b}'_3) & n_{htav}(\vec{b}'_3) & n_{htsd}(\vec{b}'_3) & n_{tkav}(\vec{b}'_3) & n_{tksd}(\vec{b}'_3) \\ n_{horz}(\vec{b}'_4) & n_{vert}(\vec{b}'_4) & n_{diag}(\vec{b}'_4) & n_{perp}(\vec{b}'_4) & n_{htav}(\vec{b}'_4) & n_{htsd}(\vec{b}'_4) & n_{tkav}(\vec{b}'_4) & n_{tksd}(\vec{b}'_4) \\ n_{horz}(\vec{b}'_5) & n_{vert}(\vec{b}'_5) & n_{diag}(\vec{b}'_5) & n_{perp}(\vec{b}'_5) & n_{htav}(\vec{b}'_5) & n_{htsd}(\vec{b}'_5) & n_{tkav}(\vec{b}'_5) & n_{tksd}(\vec{b}'_5) \\ n_{horz}(\vec{b}'_6) & n_{vert}(\vec{b}'_6) & n_{diag}(\vec{b}'_6) & n_{perp}(\vec{b}'_6) & n_{htav}(\vec{b}'_6) & n_{htsd}(\vec{b}'_6) & n_{tkav}(\vec{b}'_6) & n_{tksd}(\vec{b}'_6) \\ n_{horz}(\vec{b}'_7) & n_{vert}(\vec{b}'_7) & n_{diag}(\vec{b}'_7) & n_{perp}(\vec{b}'_7) & n_{htav}(\vec{b}'_7) & n_{htsd}(\vec{b}'_7) & n_{tkav}(\vec{b}'_7) & n_{tksd}(\vec{b}'_7) \\ n_{horz}(\vec{b}'_8) & n_{vert}(\vec{b}'_8) & n_{diag}(\vec{b}'_8) & n_{perp}(\vec{b}'_8) & n_{htav}(\vec{b}'_8) & n_{htsd}(\vec{b}'_8) & n_{tkav}(\vec{b}'_8) & n_{tksd}(\vec{b}'_8) \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ n_{horz}(\vec{b}'_q) & n_{vert}(\vec{b}'_q) & n_{diag}(\vec{b}'_q) & n_{perp}(\vec{b}'_q) & n_{htav}(\vec{b}'_q) & n_{htsd}(\vec{b}'_q) & n_{tkav}(\vec{b}'_q) & n_{tksd}(\vec{b}'_q) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{bmatrix} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \\ l_7 \\ l_8 \\ \dots \\ l_q \end{bmatrix}$$

$$\iff N\vec{a} = \vec{l}, \tag{33}$$

where N is the $q \times 8$ feature matrix, $\vec{a}$ is the correction coefficient 8-vector to solve for, and $\vec{l}$ is the length estimate error q-vector $[..., (\mathcal{L} - L_{CS}(\vec{b}'_i)), ...]$, where $i = 1, ..., q$. The model has the analytic solution

$$\vec{a} = (N^T N)^{-1} N^T \vec{l}, \tag{34}$$

using the Tikhonov regularization of $(N^T N + \lambda I)^{-1}$, where $\lambda = 1 \times 10^{-5}$, in the case where no inverse exists for $(N^T N)$. Then each $\vec{b}' \in \mathcal{B}'$ obtains its final estimate, $\mathcal{L}'$, from the correction function

$$\begin{aligned} C : \mathcal{B}' &\to \mathbb{R} \\ : \vec{b}' &\mapsto \\ a_1 n_{horz}(\vec{b}') + a_2 n_{vert}(\vec{b}') + a_3 n_{diag}(\vec{b}') + a_4 n_{perp}(\vec{b}') + \\ a_5 n_{htav}(\vec{b}') + a_6 n_{htsd}(\vec{b}') + a_7 n_{tkav}(\vec{b}') + a_8 n_{tksd}(\vec{b}'), \end{aligned} \tag{35}$$

and is given by

$$\mathcal{L}'(\vec{b}') = L_{CS}(\vec{b}') + C(\vec{b}'). \tag{36}$$

3.1.3 Linear 6-feature

We were next suspicious that the n_{htsd} and n_{tksd} standard deviation feature terms of the Linear 8-feature model were responsible for overfitting, so we created a linear 6-feature one to exclude them.

We trained a linear regression model on $q \geq 6$ calibrating molecule backbones, $\vec{b}' \in \mathcal{B}'$, having known theoretical length $\mathcal{L}$, using values from 6 of the 8 features defined above: $\{n_{horz}, n_{vert}, n_{diag}, n_{perp}, n_{htav}, n_{tkav}\}$, giving

$$\begin{bmatrix} n_{horz}(\vec{b}'_1) & n_{vert}(\vec{b}'_1) & n_{diag}(\vec{b}'_1) & n_{perp}(\vec{b}'_1) & n_{htav}(\vec{b}'_1) & n_{tkav}(\vec{b}'_1) \\ n_{horz}(\vec{b}'_2) & n_{vert}(\vec{b}'_2) & n_{diag}(\vec{b}'_2) & n_{perp}(\vec{b}'_2) & n_{htav}(\vec{b}'_2) & n_{tkav}(\vec{b}'_2) \\ n_{horz}(\vec{b}'_3) & n_{vert}(\vec{b}'_3) & n_{diag}(\vec{b}'_3) & n_{perp}(\vec{b}'_3) & n_{htav}(\vec{b}'_3) & n_{tkav}(\vec{b}'_3) \\ n_{horz}(\vec{b}'_4) & n_{vert}(\vec{b}'_4) & n_{diag}(\vec{b}'_4) & n_{perp}(\vec{b}'_4) & n_{htav}(\vec{b}'_4) & n_{tkav}(\vec{b}'_4) \\ n_{horz}(\vec{b}'_5) & n_{vert}(\vec{b}'_5) & n_{diag}(\vec{b}'_5) & n_{perp}(\vec{b}'_5) & n_{htav}(\vec{b}'_5) & n_{tkav}(\vec{b}'_5) \\ n_{horz}(\vec{b}'_6) & n_{vert}(\vec{b}'_6) & n_{diag}(\vec{b}'_6) & n_{perp}(\vec{b}'_6) & n_{htav}(\vec{b}'_6) & n_{tkav}(\vec{b}'_6) \\ ... & ... & ... & ... & ... & ... \\ n_{horz}(\vec{b}'_q) & n_{vert}(\vec{b}'_q) & n_{diag}(\vec{b}'_q) & n_{perp}(\vec{b}'_q) & n_{htav}(\vec{b}'_q) & n_{tkav}(\vec{b}'_q) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \\ ... \\ l_q \end{bmatrix} \tag{37}$$
$$\iff N\vec{a} = \vec{l},$$

where N is the $q \times 6$ feature matrix, $\vec{a}$ is the correction coefficient 6-vector to solve for, and $\vec{l}$ is the length estimate error q-vector $[..., (\mathcal{L} - L_{CS}(\vec{b}'_i)), ...]$, where $i = 1, ..., q$. The model has the analytic solution

$$\vec{a} = (N^T N)^{-1} N^T \vec{l}, \tag{38}$$

using the Tikhonov regularization of $(N^T N + \lambda I)^{-1}$, where $\lambda = 1 \times 10^{-5}$, in the case where no inverse exists for $(N^T N)$. Then each $\vec{b}' \in \mathcal{B}'$ obtains its final estimate, $\mathcal{L}'$, from the correction function

$$\begin{aligned} C : \mathcal{B}' &\to \mathbb{R} \\ : \vec{b}' &\mapsto \\ a_1 n_{horz}(\vec{b}') + a_2 n_{vert}(\vec{b}') + a_3 n_{diag}(\vec{b}') + \\ a_4 n_{perp}(\vec{b}') + a_5 n_{htav}(\vec{b}') + a_6 n_{tkav}(\vec{b}'), \end{aligned} \tag{39}$$

and is given by

$$\mathcal{L}'(\vec{b}') = L_{CS}(\vec{b}') + C(\vec{b}'). \tag{40}$$

3.2 Admissible model

3.2.1 Linear 5-feature

We were lastly suspicious that the n_{htav} term of the Linear 6-feature model was responsible for overfitting, so we created a linear 5-feature one to exclude it, and found it did not overfit the training data. This model was defined in Section 2 (Methods). Although we implicated the n_{htav} feature in the overfitting, we are not clear why this happened, but continue investigating.

3.3 Experiments

Our experiments had three datasets:

1. 19 images comprising a monodisperse set of 1,034 admissible ($L_{LS} \in [70, 100]$ nm) cDNAs having known theoretical length 75 nm
2. 44 images comprising a monodisperse set of 15,477 admissible ($L_{LS} \in [10, 1000]$ nm) cDNAs having unknown theoretical length (hereafter "Test Unknowns A")
3. 101 images comprising a monodisperse set of 54,093 admissible ($L_{LS} \in [10, 1000]$ nm) cDNAs having unknown theoretical length (hereafter "Test Unknowns B")

We partitioned the first dataset (Table 2) into the first 5 images for training (263 admissible cDNAs, $L_{LS} \in [70, 100]$, hereafter "Train") and the last 14 images for testing (2,452 admissible cDNAs, $L_{LS} \in [10, 1000]$, of which 771 have $L_{LS} \in [70, 100]$, hereafter "Test Knowns") — an arbitrary choice (see discussion below). Upon acquiring L_{CS} and the 5-feature vector, $\vec{n}$, for each of the 263 Train backbones, we trained our linear regression model by solving for the feature correction coefficients, $\vec{a}$. These are given in the "a_i" row of Table 3. We computed in-sample training residual error by plotting the histogram of $|\mathcal{L} - \mathcal{L}'_{train}|$ (Figure 8a). We then tested our $\mathcal{L}'_{train}$ estimator on the Test Knowns, Test Unknowns A, and Test Unknowns B datasets. The respective residual errors for these are the plotted histograms in Figures 9a, 10a, and 11a.

Following this, we applied spherical James-Stein shrinkage to $\vec{a}$. The spherical shrinkage factors and resulting coefficients are given in the "spherical" and "$\delta_i(\vec{a})$" rows of Table 3. We computed in-sample training residual error by plotting the histogram of $|\mathcal{L} - \mathcal{L}'_{spherical}|$ (Figure 8b). We then tested our $\mathcal{L}'_{spherical}$ estimator on the Test Knowns, Test Unknowns A, and Test Unknowns B datasets. The respective residual errors for these are the plotted histograms in Figures 9b, 10b, and 11b.

Lastly, we applied truncated James-Stein shrinkage to $\vec{a}$ using $f = 5, 4, 3$. The truncated shrinkage factors and resulting coefficients are given in the "truncated (f = 5, 4, 3)" and "$\delta_i^{(5,4,3)}(\vec{a})$" rows of Table 3. Since the truncated shrinkage factors were all so close to 1 (within 10^{-4}), they gave estimators that were essentially the trained model without shrinkage, and so we did not plot residual error histograms for these.

The feature and length estimation data for Train, Test Knowns, Test Unknowns A, and Test Unknowns B are summarized in Table 4. Note that these are the values for all admissible cDNAs. For example, the Test Unknowns A mean value of $\mathcal{L}'_{train}$ is 18.52 nm, as one might expect from a set of 15,477 alleged cDNAs in the range [10, 1000] nm, most of which are short noisy objects. To

Table 2: Number of admissible cDNAs in two $L_{LS} \in [min, max]$ nm ranges for 19 training images.

Image	[70, 100] nm	[10, 1000] nm
1	47	207
2	53	199
3	50	219
4	55	227
5	58	207
6	50	241
7	44	222
8	46	234
9	43	219
10	47	242
11	57	145
12	56	143
13	52	142
14	57	138
15	63	138
16	61	151
17	66	147
18	67	149
19	62	141

Table 3: Shrinkage factors and resulting feature correction coefficients for the Linear 5-feature model.

i	1	2	3	4	5
train	1.000000	1.000000	1.000000	1.000000	1.000000
a_i	-0.258699	-0.316009	-0.197179	-0.742293	1.637360
spherical	0.997422	0.997422	0.997422	0.997422	0.997422
$\delta_i(\vec{a})$	-0.258037	-0.315201	-0.196675	-0.740394	1.633170
truncated ($f = 5$)	0.997422	0.997422	0.997422	0.997422	0.997422
$\delta_i^{(5)}(\vec{a})$	-0.258037	-0.315201	-0.196675	-0.740394	1.633170
truncated ($f = 4$)	0.999108	0.999108	0.999108	0.999108	0.999596
$\delta_i^{(4)}(\vec{a})$	-0.258468	-0.315727	-0.197003	-0.741631	1.636700
truncated ($f = 3$)	0.999655	0.999655	0.999655	0.999853	0.999933
$\delta_i^{(3)}(\vec{a})$	-0.258610	-0.315900	-0.197111	-0.742184	1.637250

focus on the areas of interest, around the theoretical known lengths of the data, we produced the feature and length estimation data for Train, Test Knowns, Test Unknowns A, and Test Unknowns B summarized in Table 5. Note that this is a small subset of the data summarized in Table 4. Here the selection criterion was $\mathcal{L}' \in [\mathcal{L} - 15, \mathcal{L} + 15]$ nm. Our motivation was to evaluate the mean and standard deviation of $\mathcal{L}'$ estimated lengths in the band 2σ nm away from the theoretical known lengths of each monodisperse dataset. We assumed $\sigma = 7.5$ nm from our problem statement. The theoretical known lengths of Test Unknowns A (223 nm) and Test Unknowns B (584 nm) were revealed to us after our experiments by our collaborator who provided us with the image data, allowing us to perform this retrospective analysis.

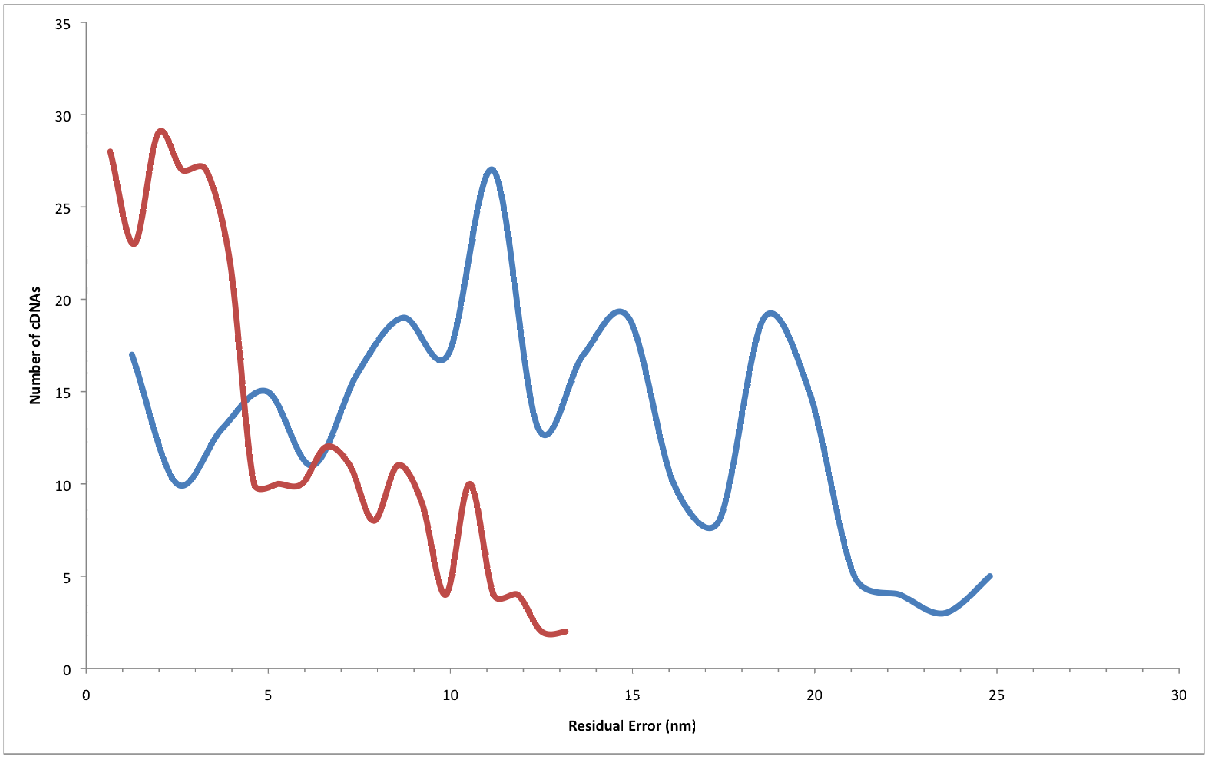
The residual error histograms showed that for each test dataset, there was clearly one high frequency area of interest, near zero (i.e. when the $\mathcal{L}'$ estimates were close to the known theoretical lengths, $\mathcal{L}$, of the monodisperse cDNAs in each test dataset). Looking at the 2σ band about $\mathcal{L}$ in Table 5, we found $\mathcal{L}'_{train}$ had mean values of: 74.62 nm (5.29 nm SD) for Train, 76.95 nm (7.10 nm SD) for Test Knowns, 229.22 nm (6.03 nm SD) for Test Unknowns A, and 578.77 nm (7.57 nm SD) for Test Unknowns B — all within the tolerance of the problem statement. The mean and SD results for $\mathcal{L}'_{spherical}$ and the $f = 5, 4, 3$ versions of $\mathcal{L}'_{truncated}$ gave no improvement over $\mathcal{L}'_{train}$ (as evinced by their shrinkage factors being so close to 1, in Table 3). i.e. The James-Stein shrinkage process suggested our Linear 5-feature model was a good fit, and not an overfit, to the training data.

Table 4: Admissible feature and length estimation results for the Linear 5-feature model.

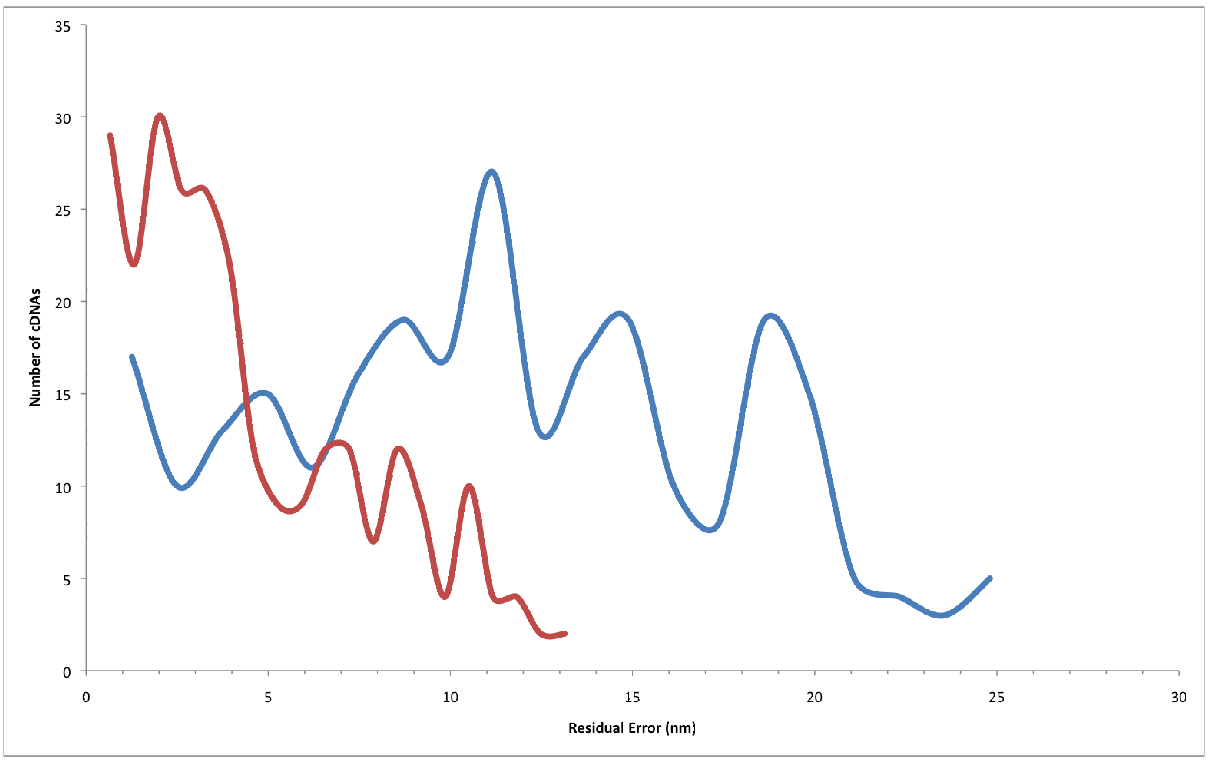
	Train Knowns $\mathcal{L} = 75$ nm 0.97 $\frac{nm}{pixel}$ $L_{LS} \in [70, 100]$ nm 5 images 263 cDNAs		Test Knowns $\mathcal{L} = 75$ nm 0.97 $\frac{nm}{pixel}$ $L_{LS} \in [10, 1000]$ nm 14 images 2,452 cDNAs		Test Unknowns A $\mathcal{L} = 223$ nm 0.97 $\frac{nm}{pixel}$ $L_{LS} \in [10, 1000]$ nm 44 images 15,477 cDNAs		Test Unknowns B $\mathcal{L} = 584$ nm 0.97 $\frac{nm}{pixel}$ $L_{LS} \in [10, 1000]$ nm 101 images 54,093 cDNAs	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
n_{horz}	18.74	11.16	12.87	11.60	5.64	7.84	6.53	12.45
n_{vert}	21.30	12.40	14.44	13.47	2.84	6.33	3.97	16.34
n_{diag}	33.55	5.22	22.90	16.57	7.36	10.28	8.48	21.10
n_{perp}	1.40	1.31	1.07	1.22	0.77	1.14	0.75	1.48
n_{htav}	242.67	2.29	242.73	3.85	239.33	3.75	240.33	4.61
n_{htsd}	4.92	0.84	4.63	1.20	4.24	1.41	4.48	1.58
n_{tkav}	5.11	1.31	5.45	2.08	2.73	2.06	3.22	2.09
n_{tksd}	1.82	0.73	2.02	0.94	1.69	0.75	1.76	0.75
L_{LS}	85.45	6.71	58.31	40.77	18.45	26.29	21.98	54.12
L_{CS}	85.48	6.72	58.32	40.78	18.44	26.30	21.97	54.14
$\mathcal{L}'_{train}$	74.62	5.29	54.04	33.18	18.52	22.50	22.07	43.14
$\mathcal{L}'_{spherical}$	74.65	5.29	54.05	33.20	18.52	22.51	22.07	43.16
$\mathcal{L}'_{truncated}(f = 5)$	74.65	5.29	54.05	33.20	18.52	22.51	22.07	43.16
$\mathcal{L}'_{truncated}(f = 4)$	74.63	5.29	54.05	33.19	18.52	22.51	22.07	43.15
$\mathcal{L}'_{truncated}(f = 3)$	74.63	5.29	54.05	33.19	18.52	22.51	22.07	43.14

Table 5: Admissible feature and length estimation results for the Linear 5-feature model, a subset view within $2\sigma = 15$ nm of the theoretical known lengths.

	Train Knowns $\mathcal{L} = 75$ nm 0.97 $\frac{nm}{pixel}$ $L_{LS} \in [70, 100]$ nm $\mathcal{L}' \in [75 \pm 15]$ nm 5 images 263 cDNAs		Test Knowns $\mathcal{L} = 75$ nm 0.97 $\frac{nm}{pixel}$ $L_{LS} \in [10, 1000]$ nm $\mathcal{L}' \in [75 \pm 15]$ nm 14 images 860 cDNAs		Test Unknowns A $\mathcal{L} = 223$ nm 0.97 $\frac{nm}{pixel}$ $L_{LS} \in [10, 1000]$ nm $\mathcal{L}' \in [223 \pm 15]$ nm 44 images 74 cDNAs		Test Unknowns B $\mathcal{L} = 584$ nm 0.97 $\frac{nm}{pixel}$ $L_{LS} \in [10, 1000]$ nm $\mathcal{L}' \in [584 \pm 15]$ nm 101 images 65 cDNAs	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
n_{horz}	18.74	11.16	19.14	9.66	74.06	19.72	113.89	76.06
n_{vert}	21.30	12.40	20.66	11.21	55.02	19.41	227.87	77.34
n_{diag}	33.55	5.22	34.11	5.93	104.66	8.82	289.09	20.12
n_{perp}	1.40	1.31	1.52	1.23	6.45	3.06	16.86	5.67
n_{htav}	242.67	2.29	244.16	2.47	250.84	0.90	252.82	0.84
n_{htsd}	4.92	0.84	4.67	0.85	4.05	0.81	2.05	0.60
n_{tkav}	5.11	1.31	6.28	1.63	12.55	1.13	10.17	1.19
n_{tksd}	1.82	0.73	2.04	0.87	3.54	0.91	2.28	0.58
L_{LS}	85.45	6.71	85.99	9.71	270.61	9.36	733.01	13.79
L_{CS}	85.48	6.72	86.01	9.70	270.64	9.34	733.10	13.74
$\mathcal{L}'_{train}$	74.62	5.29	76.95	7.10	229.22	6.03	578.77	7.57
$\mathcal{L}'_{spherical}$	74.65	5.29	76.98	7.10	229.21	5.99	578.73	7.68
$\mathcal{L}'_{truncated}(f = 5)$	74.65	5.29	76.98	7.10	229.21	5.99	578.73	7.68
$\mathcal{L}'_{truncated}(f = 4)$	74.63	5.29	76.97	7.10	229.27	6.03	578.63	7.64
$\mathcal{L}'_{truncated}(f = 3)$	74.63	5.29	76.96	7.10	229.24	6.03	578.83	7.57

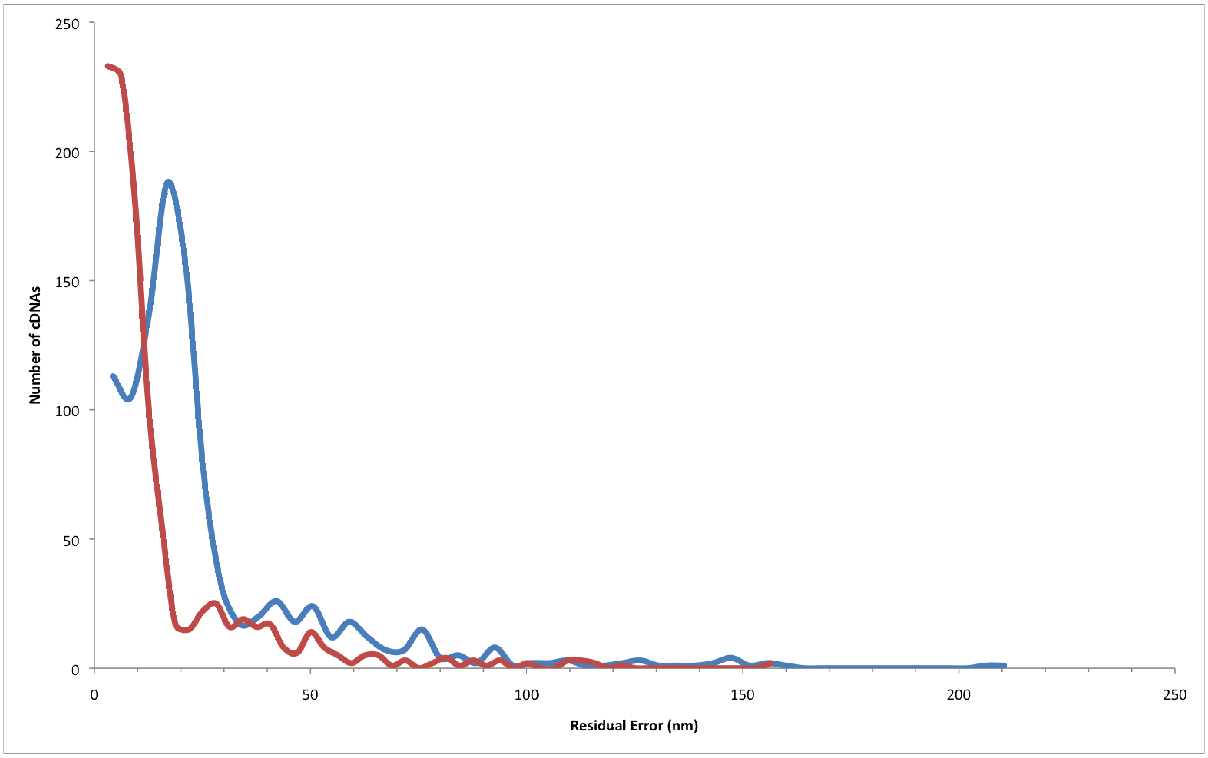


(a) $|\mathcal{L} - \mathcal{L}'_{train}|$ (red) vs $|\mathcal{L} - L_{CS}|$ (blue)

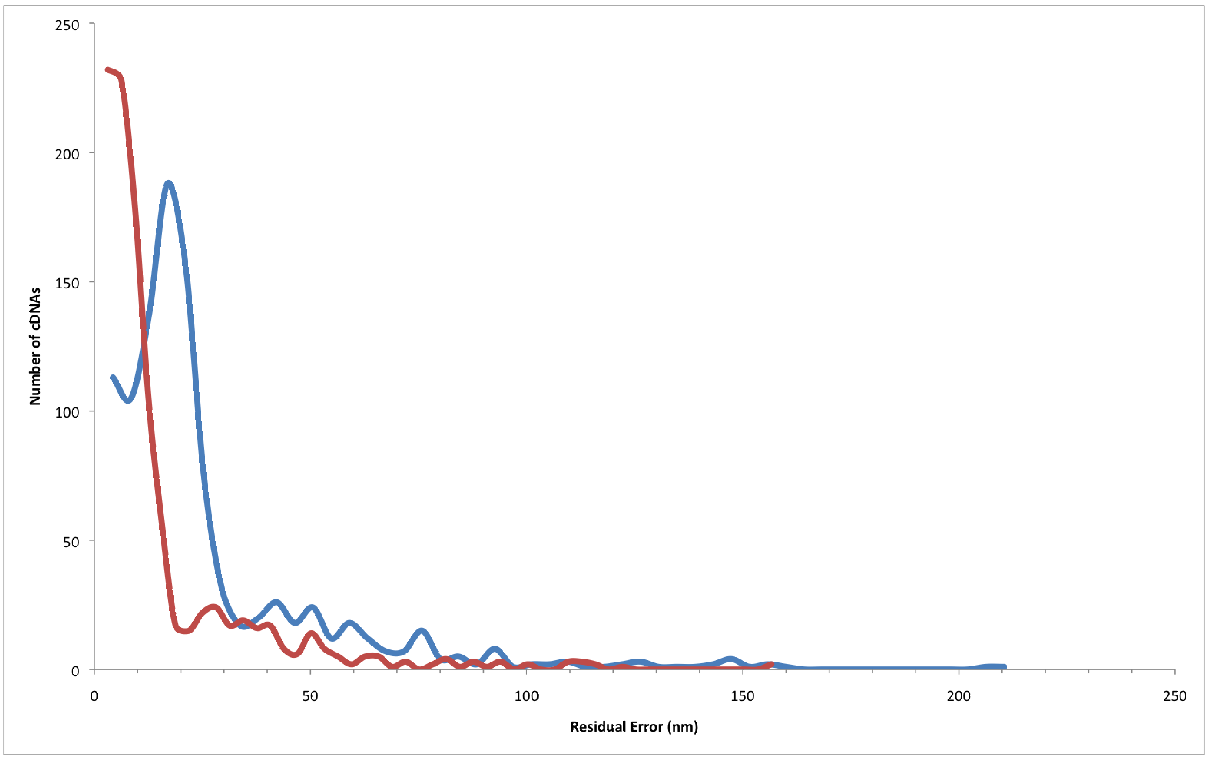


(b) $|\mathcal{L} - \mathcal{L}'_{spherical}|$ (red) vs $|\mathcal{L} - L_{CS}|$ (blue)

Figure 8: Residual error distributions before (8a) and after (8b) spherical shrinkage for the Linear 5-feature model upon Train.

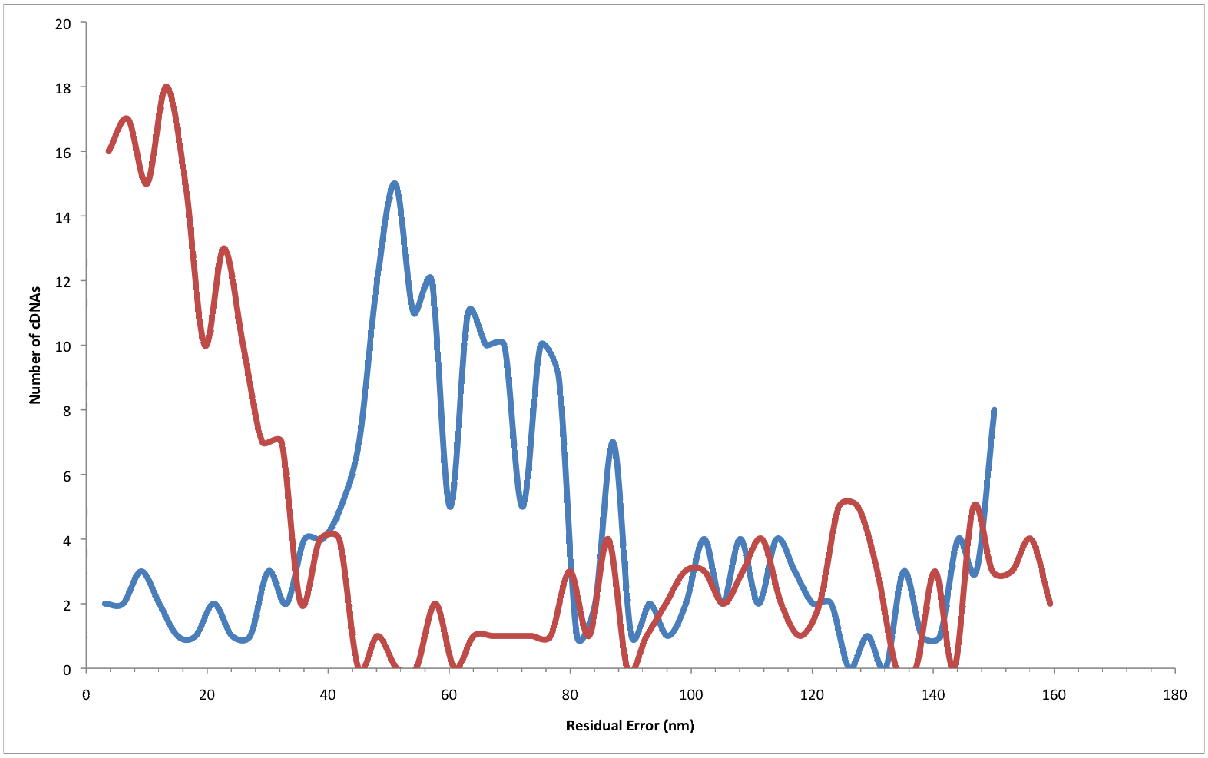


(a) $|\mathcal{L} - \mathcal{L}'_{train}|$ (red) vs $|\mathcal{L} - L_{CS}|$ (blue)

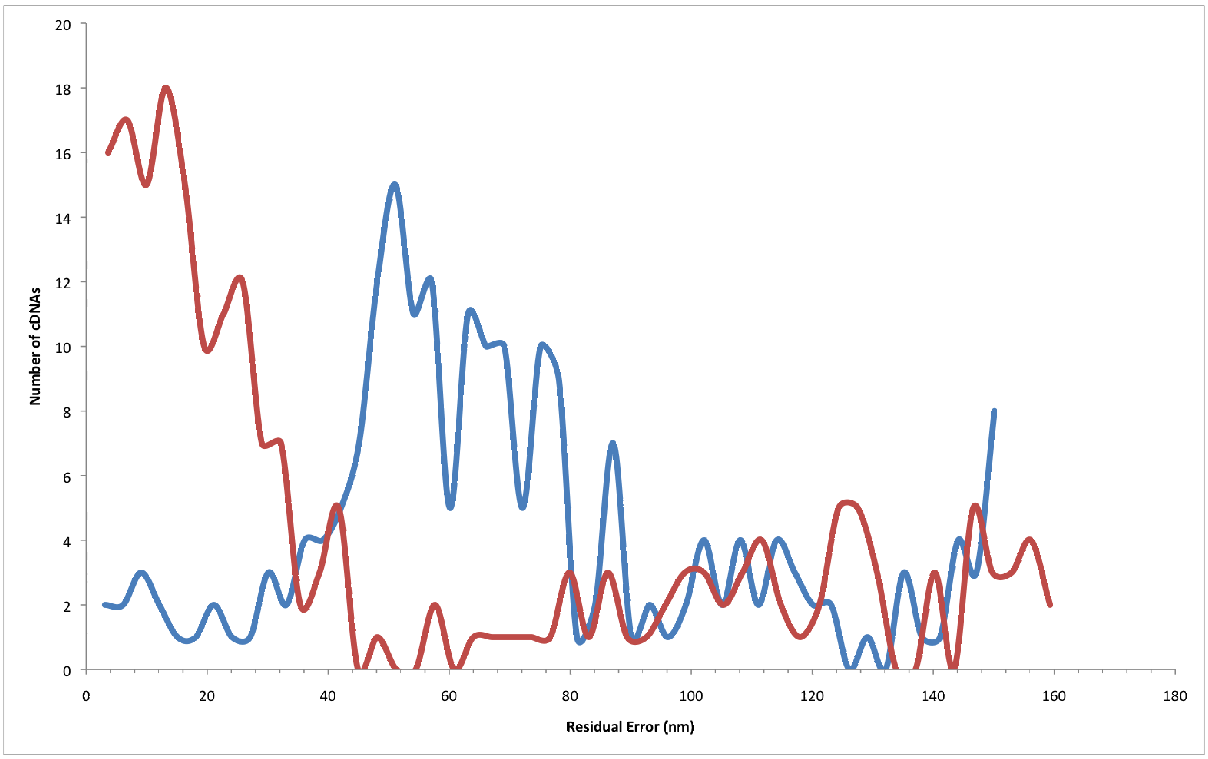


(b) $|\mathcal{L} - \mathcal{L}'_{spherical}|$ (red) vs $|\mathcal{L} - L_{CS}|$ (blue)

Figure 9: Residual error distributions before (9a) and after (9b) spherical shrinkage for the Linear 5-feature model upon Test Knowns. Although $L_{LS} \in [10, 1000]$ nm were admitted, only $L_{LS} \in [70, 1000]$ are shown here, to filter the noise of so many short objects.

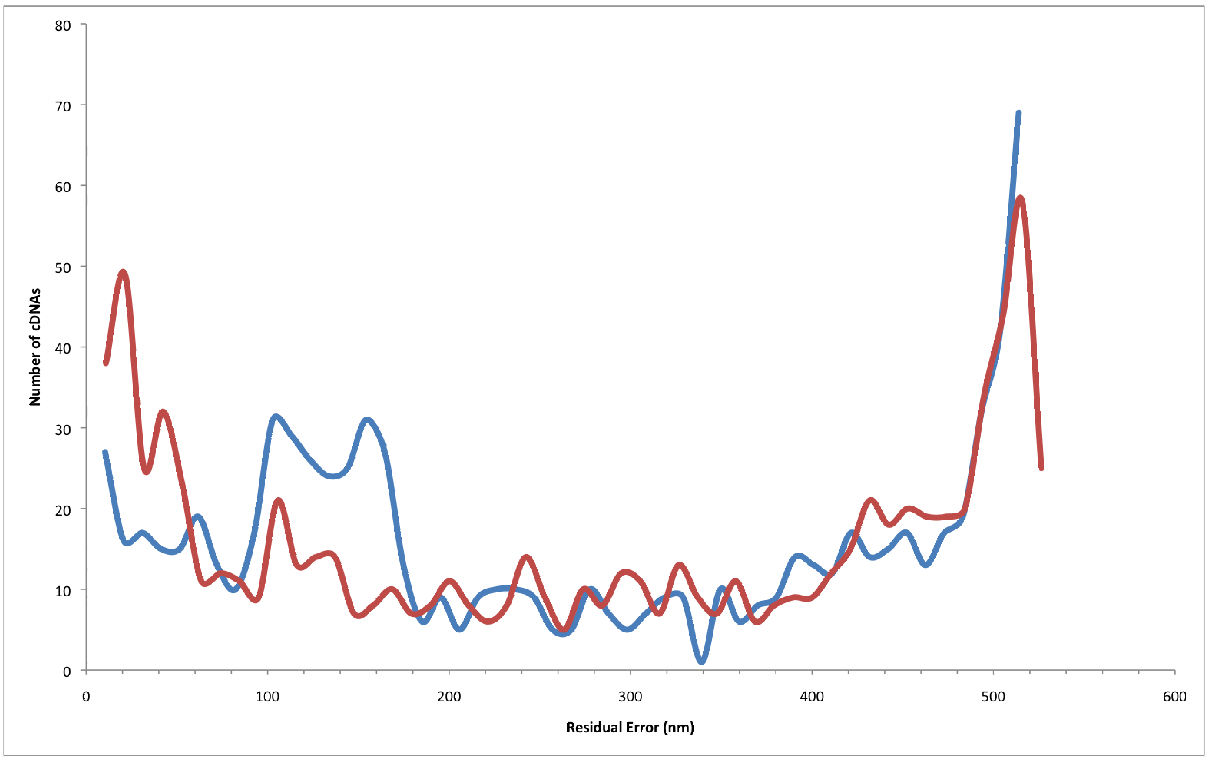


(a) $|\mathcal{L} - \mathcal{L}'_{train}|$ (red) vs $|\mathcal{L} - L_{CS}|$ (blue)

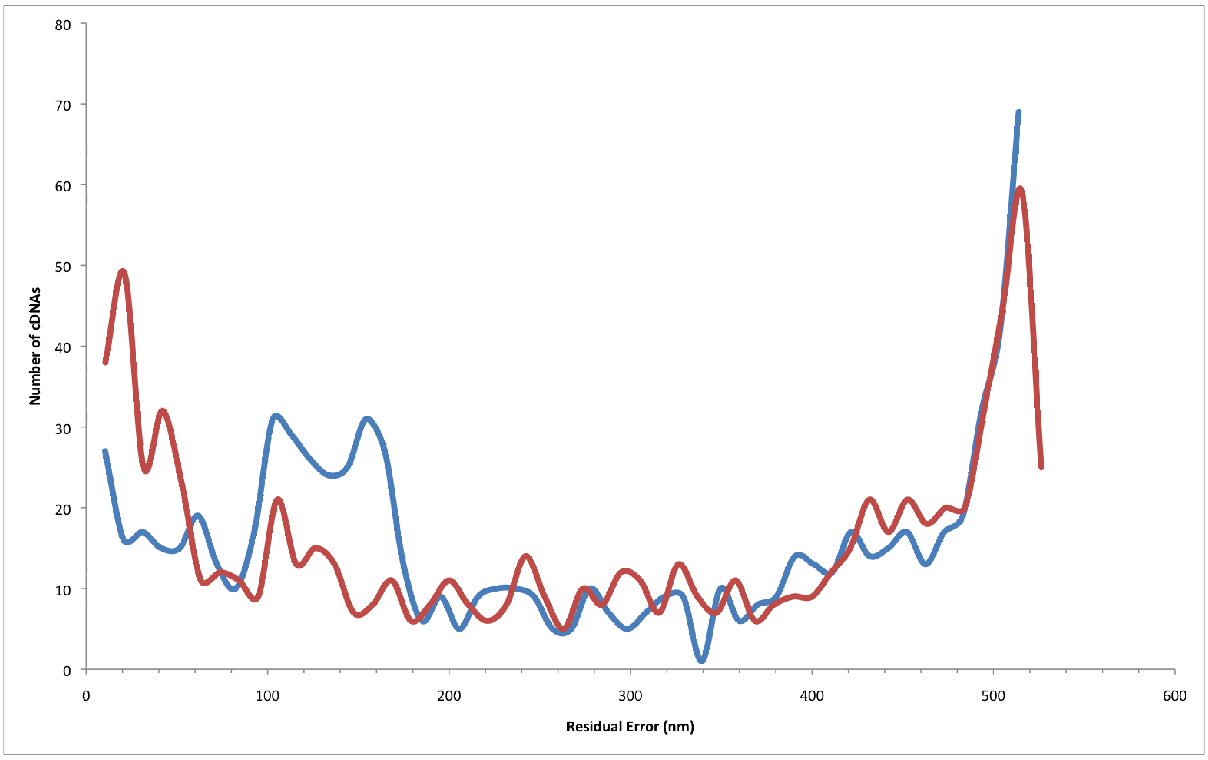


(b) $|\mathcal{L} - \mathcal{L}'_{spherical}|$ (red) vs $|\mathcal{L} - L_{CS}|$ (blue)

Figure 10: Residual error distributions before (10a) and after (10b) spherical shrinkage for the Linear 5-feature model upon Test Unknowns A. Although $L_{LS} \in [10, 1000]$ nm were admitted, only $L_{LS} \in [70, 1000]$ are shown here, to filter the noise of so many short objects.



(a) $|\mathcal{L} - \mathcal{L}'_{train}|$ (red) vs $|\mathcal{L} - L_{CS}|$ (blue)



(b) $|\mathcal{L} - \mathcal{L}'_{spherical}|$ (red) vs $|\mathcal{L} - L_{CS}|$ (blue)

Figure 11: Residual error distributions before (11a) and after (11b) spherical shrinkage for the Linear 5-feature model upon Test Unknowns B. Although $L_{LS} \in [10, 1000]$ nm were admitted, only $L_{LS} \in [70, 1000]$ are shown here, to filter the noise of so many short objects. Note, however, the apparent bimodal distribution. We suspect the Unknowns B dataset is not truly monodisperse.

4 Discussion

4.1 Main Message

In this problem, there are two principal sources of error: bias from the method of estimation (the extrinsic factors), and systematic error (the intrinsic factors). We have given a BLUE estimator for molecular backbone contour length, namely the piecewise cubic spline fitting measure, L_{CS}. But this gets us only part way to the answer, since systematic error underlies all such measurements. In our experiments, this was visible in the residual error distributions (Figures 8, 9, 10, and 11), especially in Test Unknowns A and Test Unknowns B, where the divergence between L_{CS} and $\mathcal{L}'$ was clear. Various sources of systematic error are discussed below.

Accordingly, we were motivated by the following claim:

CLAIM 3. *Systematic error in this problem can be characterized by a set of detectable and measurable image features whose values depend on the length of molecular backbone contours. i.e. Systematic error can be estimated from the data by training the correct model.*

We improved on L_{CS} by training a linear regression model to estimate the systematic error and thereby correct L_{CS}, yielding a superior estimator, $\mathcal{L}'_{train}$. This was trained on 5 of the 8 features we developed for our method. Further, we developed a James-Stein shrinkage procedure, in spherical and truncated forms (giving potentially further improved estimators, $\mathcal{L}'_{spherical}$ and $\mathcal{L}'_{truncated}$), that trades off bias for variance in the model coefficients, effectively throttling those introducing too much variance into the estimation. One consequence of such a design is an inherent adaptability and extensibility: a researcher may compose any number and arrangement of features into the estimation. We believe our approach will help ameliorate the model selection problem in this context.

4.2 Critical Assessment

4.2.1 Methods

We are not clear that our way of evaluating the quality of the $\mathcal{L}'$ estimators presented in Table 5 uses a proper methodology. While the $\mathcal{L}'$ estimators decisively clustered near $\mathcal{L}$ in each test dataset (giving high frequencies near zero residual error), the spread in the values outside of the $2\sigma = 15$ nm band might exceed the $\sigma = 7.5$ nm tolerance. We chose what seemed a reasonable band around the theoretical known lengths, whose $\mathcal{L}'$ values therein spread roughly half of that quantity. But this needs more work.

In terms of learning feature values from the data, we would like to explore alternatives to linear regression. For instance, AdaBoost with decision stumps. In terms of biased estimation, we would like to explore alternatives to James-Stein shrinkage. For instance, principal component regression, ridge regression, and LASSO — subjects for a follow-up study.

4.2.2 Cross validation to select model terms

In terms of the model selection problem, we could be more exhaustive to improve our results. For example, we could employ the leave-one-out cross validation method to prune model terms. Combining this with a similar scheme for data cross validation would require a large number of experiments. If a model has p terms and a test dataset has q admissible molecules, then this would require pq training and testing regimes to find the best model using the original one as a template.

4.2.3 Cross validation to measure test error

Two obvious choices for this are leave-one-out cross validation (mentioned above) and k-fold cross validation. In the crudest configuration, given our datasets (now with all three monodisperse theoretical lengths known to us), we could design the following training and testing regimes to round out our results:

1. Train on a subset of the 75 nm; test on the complimentary Train subset and the $\{223, 584\}$ nm knowns [completed and discussed here]
2. Train on a subset of the 223 nm; test on the complimentary Train subset and the $\{75, 584\}$ nm knowns
3. Train on a subset of the 584 nm; test on the complimentary Train subset and the $\{75, 223\}$ nm knowns
4. Train on a subset of the $\{75, 223\}$ nm; test on the complimentary Train subset and the 584 nm knowns
5. Train on a subset of the $\{75, 584\}$ nm; test on the complimentary Train subset and the 223 nm knowns
6. Train on a subset of the $\{223, 584\}$ nm; test on the complimentary Train subset and the 75 nm knowns
7. Train on a subset of the $\{75, 223, 584\}$ nm; test on the complimentary Train subset and any novel and truly unknown data

Of course, within each of these one can also employ k-fold cross validation.

4.3 Comparison with Other Studies

The other studies we found took the image processing aspect of the problem to the limit. The approach taken by Ficarra, *et al.* [13] is a good example. These studies also use simple length correction methods to address the errors that pixel quantization imposes upon the smooth and continuous molecular backbone contours whose lengths are to be estimated. Regarding systematic error estimation, these studies all use an image processing step to thin two-dimensional objects into one-dimensional 8-connected pixel paths, and some approaches reclaim pixels at the ends, while others argue this is unfounded. But this is as far as they go to address the tip convolution problem, discussed below; rather, they assume the dilation effects are symmetric and uniform, while this may not be the case. And none of these studies address the problem of thermal drift, discussed below.

We give a meta-approach to the problem of backbone contour length estimation that learns to characterize the systematic error from the data, namely, image features whose values depend on the lengths of backbone contours. While we do not explicitly model tip convolution and thermal drift, for example, these effects are presumably measured by the collection of features employed by the selected model.

4.4 Conclusions

Our meta-approach showed promise in our experiments, meeting our tolerance requirement, but it needs wider experimental validation, in terms of model design and variations in monodisperse test data.

5 Future Work

At this juncture, we would like to briefly mention some of the intrinsic problems that generate systematic error. These are open problems, and so present opportunities for researchers to create explicit models that will result in better (albeit computationally more complex) features to exploit in the general pursuit of modeling systematic error.

5.1 AFM data

5.1.1 Use phase shift data to quantify softness

All the approaches under review, including ours, make use of half of the AFM data available. For each point (x, y) in the area under inspection, the AFM instrument in tapping mode takes two measurements: the displacement in the z-direction for *height* (the typical AFM "image"), and the change in oscillation frequency for *softness*. Making use of this data could significantly improve the resolution of fine structures in the objects, and lead to the development of new model features.

5.2 Problems implicit to AFM

5.2.1 Tip convolution

As the AFM scans the surface, the tip makes contact with, or taps with high frequency in close proximity to, the sample being measured. The geometric interaction between tip and sample results in a profile that is a convolution of the two geometries. In AFM images of biomlecules, their ends appear longer than they should be, and their widths appear thicker than they should be. Deconvolving the image requires a special kind of algorithm, because the space of all possible tip-sample geometry combinations that can produce the same profile is large, and must be selected with care. Further complicating the problem, since tips are brittle, they get deformed in irregular ways, or simply break off. Experimental protocol should dictate tip calibration, to model its shape for post-process deconvolution.

Moreover, this is an example of the problem of Wittgenstein's Ruler:

> "Unless you have confidence in the ruler's reliability, if you use a ruler to measure a table, you may also be using the table to measure the ruler. The less you trust the ruler's reliability (in probability called the prior), the more information you are getting about the ruler and the less about the table." [36]

So is the tip measuring the sample, or is the sample measuring the tip?

None of the approaches under review, including ours, attempt to model tip convolution effects directly and appropriately deconvolve the image, though the problem is widely acknowledged [22, 38, 7, 37, 40] and algorithms designed precisely for this purpose exist [39]. In our experiments, we have consistently observed 5 nm added to each backbone end, and 5 nm added to the thickness, of biomolecules we measure.

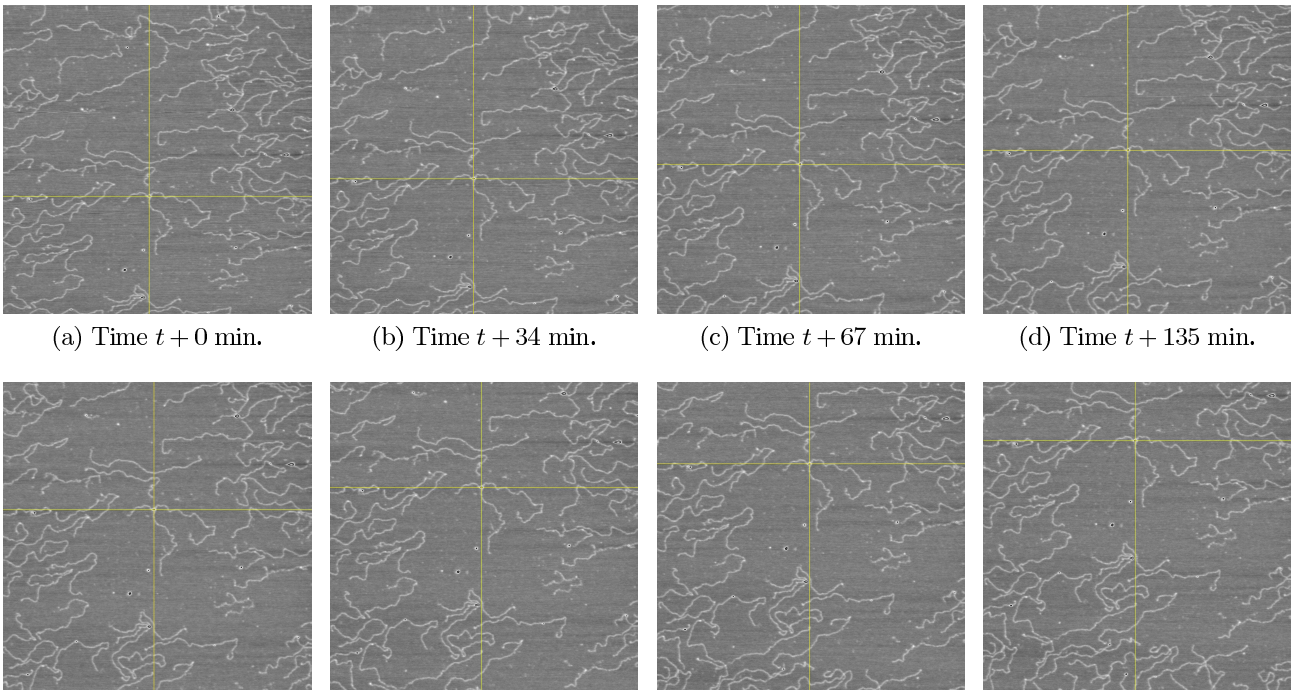
5.2.2 Thermal drift

Each component of the AFM instrument — tip, cantilever arm, photodiode, stage, piezoelectronics, sample, substrate, etc. — has its own coefficient of thermal expansion. Even minute fluctuations in ambient temperature lead to an aggregate displacement of the materials with respect to each other, and hence drift. Experimenters observe drift vectors that range from unidirectional to

random walks. Some observe the drift reaches an equilibrium within minutes, others hours. Since environmental fluctuations may occur during a single scan period (say 30 minutes), drift may vary across a single image, resulting in local rather than global distortions, and to varying degrees.

None of the approaches under review, including ours, attempt to model thermal drift directly and perform the appropriate deblurring of the image (locally or globally) though this problem too is widely acknowledged [4, 41, 27, 44] and an assortment of well-suited algorithms for this exist, namely Carasso's SECB algorithm [2, 3]. Another approach that we have considered is to use calibrating molecules as local rulers — the subject of a follow-up paper.

To illustrate and empirically quantify the thermal drift effects present in our experiments, we took 8 top-to-bottom scans in succession over a period of 404 minutes (Figure 12). Overlaying the first two scans, we saw the drift is local and nonlinear (Figure 13). The displacement vector from the first to the second scan gave a drift rate of 3.3 $\frac{nm}{min}$ (Figure 14). A more accurate average drift rate was computed from the displacement vector from the first to the last scan, which gave a drift rate of 2.1 $\frac{nm}{min}$ (Figure 15).



(a) Time $t + 0$ min. (b) Time $t + 34$ min. (c) Time $t + 67$ min. (d) Time $t + 135$ min.

(e) Time $t + 202$ min. (f) Time $t + 270$ min. (g) Time $t + 337$ min. (h) Time $t + 404$ min.

Figure 12: Thermal drift over 404 minutes in repeated AFM top-to-bottom scans of a 2000×2000 nm area. The yellow crosshairs track a single anchored position across the 8 images.

5.2.3 Non-closed-loop scanning

Experimenters can use closed-loop scanning settings in their protocols, to reduce the effects of thermal drift by spending the majority of scan time on just the objects of interest. We have observed non-closed-loop scanning to give between 2 and 5 percent error in sizing the microscopic area. Since this area measure is used to calculate the $\frac{nm}{pixel}$ ratio at the outset of experimentation, this error propagates through all downstream computations.

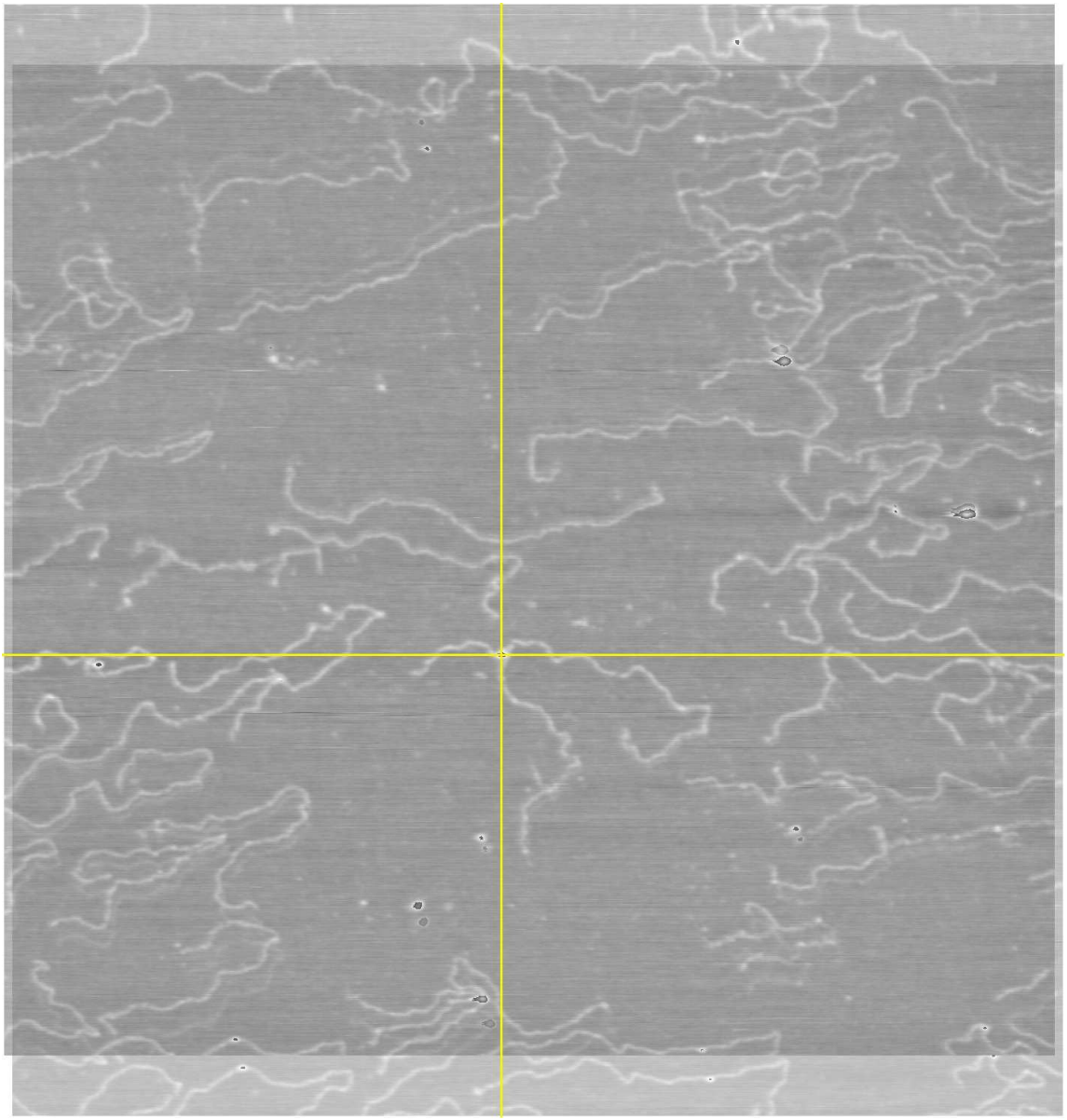


Figure 13: Alignment of the first two AFM top-to-bottom scans (at $t + 0$ min and $t + 34$ min), demonstrating the local and nonlinear character of the thermal drift

.

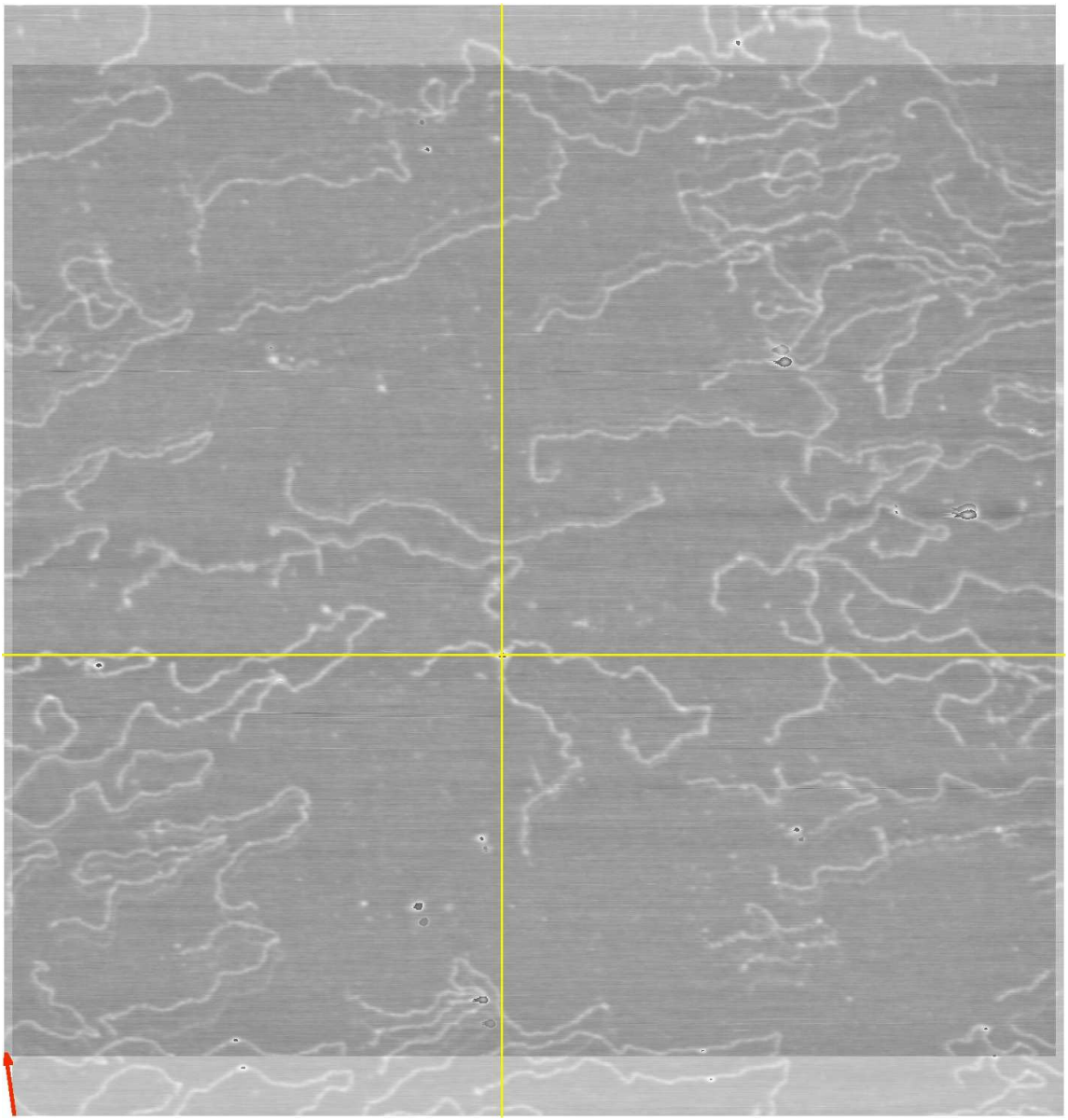


Figure 14: Alignment of the first two AFM top-to-bottom scans (at $t + 0$ min and $t + 34$ min), with displacement vector (red) added. The vector magnitude is 80 pixels, which, at 1.42 $\frac{nm}{pixel}$ resolution, gives a displacement of 113 nm, at a rate of 3.3 $\frac{nm}{min}$.

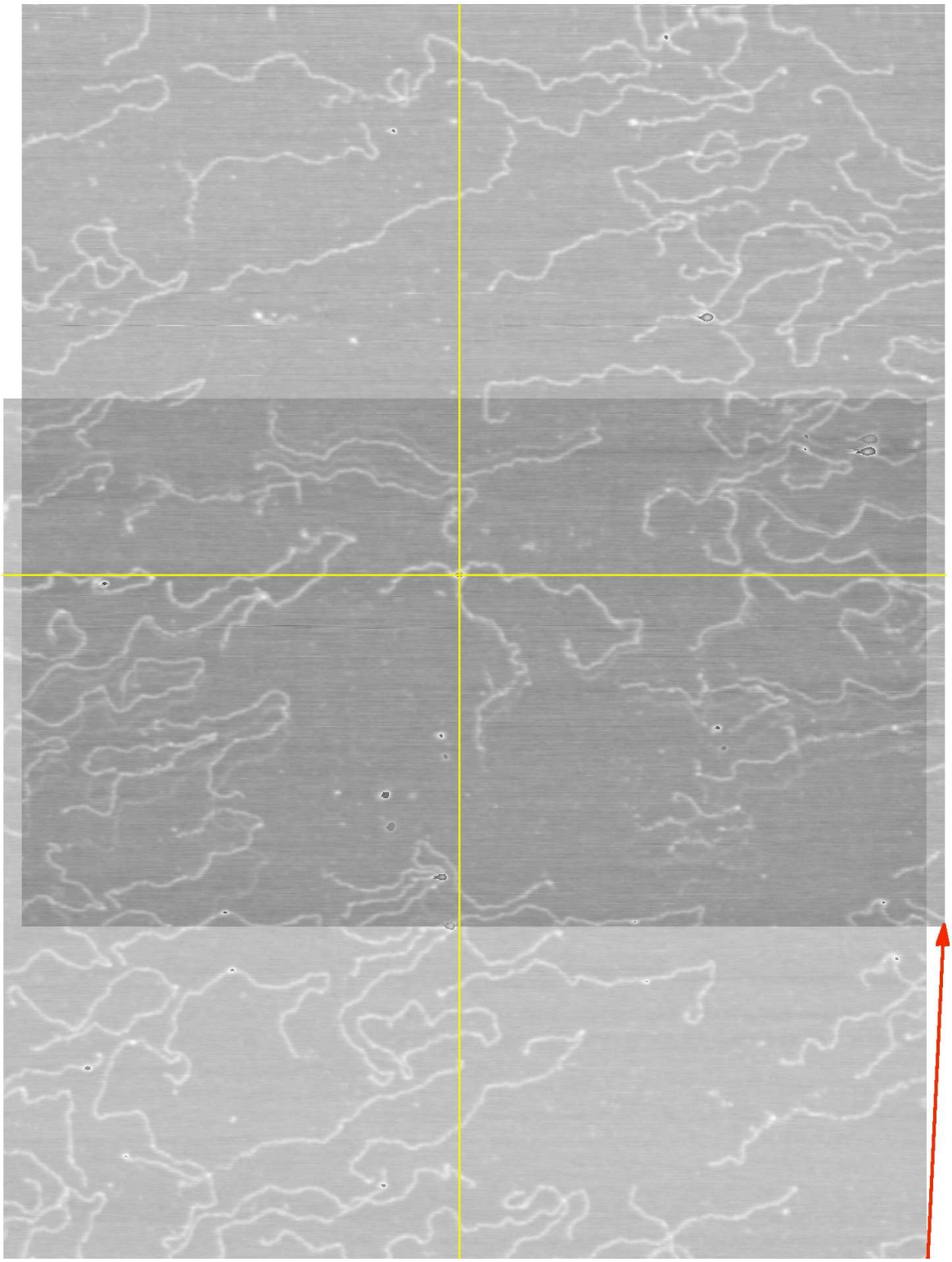


Figure 15: Alignment of the first and last AFM top-to-bottom scans (at $t + 0$ min and $t + 404$ min), with displacement vector (red) added. The vector magnitude is 593 pixels, which, at 1.42 $\frac{nm}{pixel}$ resolution, gives a displacement of 842 nm, at a rate of 2.1 $\frac{nm}{min}$, representing the average drift rate over the net displacement in 404 min.

5.3 Problems implicit to chemistry

5.3.1 3-D to 2-D conformation during adsorption

It is not well understood how a three-dimensional DNA or RNA molecule adsorbs onto a substrate like mica, and under what conditions uniform binding to the surface occurs, let alone how to ensure this. We expect better models will emerge that will eventually lead to reduction in these kinds of experimental error.

6 Acknowledgements

Bud Mishra for guidance through this research, including his principal reading of this thesis. Davi Geiger for his insightful second reading of this thesis. Jason Reed for a stimulating collaboration with his laboratory in the California NanoSystems Institute at UCLA, beginning August 2007. Members of the NYU Bioinformatics Lab for their collective encouragement, and for being an inquisitive and critical audience for this research during its presentation.

References

[1] H. Beffert and R. Shinghal. Skeletonizing binary patterns on the homogeneous multiprocessor. *Intl. J. Patt. Reco. Art. Intell.*, 3:2:207–216, 1989.

[2] A.S. Carasso. Error bounds in nonsmooth image deblurring. *SIAM J. Math. Anal.*, 28:3:656–668, 1997.

[3] A.S. Carasso. Linear and nonlinear image deblurring: A documented study. *SIAM J. Numer. Anal.*, 36:6:1659–1689, 1999.

[4] Y. Chen and W. Huang. Application of a novel nonperiodic grating in scanning probe microscopy drift measurement. *Rev. Sci. Instr.*, 78:7, 2007.

[5] S. Cirrone. Riconoscimento automatico ed analisi di molecole di DNA mediante elaborazione di immagini AFM. [Automatic recognition and analysis of DNA molecules by AFM image processing.] (In Italian.). Master's thesis, Universitá degli Studi di Catania Facoltá di Ingegneria, 2007.

[6] D. Coeurjolly and R. Klette. A comparative evaluation of length estimators of digital curves. *IEEE Trans. Patt. Anal. Mach. Intel.*, 26:2:252–258, 2004.

[7] G. Dahlen, M. Osborn, N. Okulan, W. Foreman, and A. Chand. Tip characterization and surface reconstruction of complex structures with critical dimension atomic force microscopy. *J. Vac. Sci. Technol. B*, 23:6:2297–2303, 2005.

[8] D. Dey and J.O. Berger. On truncation of shrinkage estimators in simultaneous estimation of normal means. *J. Amer. Stat. Assoc.*, 78:384:865–869, 1983.

[9] L. Dorst and A.W.M. Smeulders. Length estimators for digitized contours. *Comp. Vis. Graph. Image Proc.*, 40:311–333, 1987.

[10] L. Dorst and A.W.M. Smeulders. Discrete straight line segments: Parameters, primitives and properties. In *Vision Geometry, series Contemporary Mathematics*, pages 45–62. American Mathematical Society, 1991.

[11] Y. Fang, T.S. Spisz, T. Wiltshire, N.P. D'Costa, I.N. Bankman, R.H. Reeves, and J.H. Hoh. Solid-state DNA sizing by atomic force microscopy. *Anal. Chem.*, 70:10:2123–2129, 1998.

[12] G. Feigin and N. Ben-Yosef. Line thinning algorithm. In *SPIE Proceedings Series V: Applications of Digital Image Processing*, volume 397, page 108, 1983.

[13] E. Ficarra, L. Benini, E. Macii, and G. Zuccheri. Automated DNA fragments recognition and sizing through AFM image processing. *IEEE Trans. Info. Technol. Biomed.*, 9:4:508–517, 2005.

[14] E. Ficarra, L. Benini, B. Ricco, and G. Zuccheri. Automated DNA sizing in atomic force microscope images. *IEEE Intl. Symp. on Biomed. Imaging*, 17:10:30.0:453–456, 2002.

[15] E. Ficarra, E. Macii, L. Benini, and G. Zuccheri. A robust algorithm for automated analysis of DNA molecules in AFM images. In *Proc. Biomed. Eng.*, volume 417, 2004.

[16] E. Ficarra, D. Masotti, L. Benini, M. Milano, and A. Bergia. A robust algorithm for automated analysis of DNA molecules in AFM images. *AI*IA Notizie*, 4:64–68, 2002.

[17] E. Ficarra, D. Masotti, E. Macii, L. Benini, and B. Samori. Automatic intrinsic DNA curvature computation from AFM images. *IEEE Trans. Biomed. Eng.*, 52:12:2074–2086, 2005.

[18] M.A.T. Figueiredo, J.M.N. Leitão, and A.K. Jain. Unsupervised contour representation and estimation using B-splines and a minimum description length criterion. *IEEE Trans. Image Proc.*, 9:6:1075–1087, 2000.

[19] H. Freeman. Techniques for the digital computer analysis of chain-encoded arbitrary plane curves. In *Proc. Nat'l. Elec. Conf.*, volume 17, pages 421–432, 1961.

[20] W. James and C. Stein. Estimation with quadratic loss. In *Proc. Berkeley Symp. Math. Stat. Prob.*, pages 316–379, 1961.

[21] V. Kalmykov. Structural analysis of contours as the sequences of the digital straight segments and of the digital curve arcs. *Intl. J. Info. Th. Appl.*, 14:3:238–243, 2007.

[22] D. Keller. Reconstruction of STM and AFM images distorted by finite-size tips. *Surf. Sci.*, 253:353–364, 1991.

[23] R. Klette, V. Kovalevsky, and B. Yip. On the length estimation of digital curves. Technical report, University of Auckland, May 1999. CITR-TR-45.

[24] L. Lam, S-W. Lee, and C.Y. Suen. Thinning methodologies — a comprehensive survey. *IEEE Trans. Patt. Anal. Mach. Intel.*, 14:9:869–885, 1992.

[25] R. Marcondes Cesar Jr. and L. da Fontoura Costa. Towards effective planar shape representation with multiscale digital curvature analysis based on signal processing techniques. *Patt. Recog.*, 29:1559–1569, 1996.

[26] J. Marek, E. Demjénová, Z. Tomori, J. Janáček, I. Zolotová, F. Valle, M. Favre, and G. Dietler. Interactive measurement and characterization of DNA molecules by analysis of AFM images. *Cytometry*, 63A:2:87–93, 2005.

[27] B. Mokaberi and A.A.G. Requicha. Towards automatic nanomanipulation: Drift compensation in scanning probe microscopes. In *Proc. IEEE Intl. Conf. Rob. Automat.*, volume 1, pages 416–421, 2004.

[28] J. Reed, B. Mishra, B. Pittenger, S. Magonov, J. Troke, M.A. Teitell, and J.K. Gimzewski. Single molecule transcription profiling with AFM. *Nanotechnology*, 18:4:1–15, 2007.

[29] C. Rivetti and S. Codeluppi. Accurate length determination of DNA molecules visualized by atomic force microscopy: Evidence for a partial b- to a-form transition on mica. *Ultramicroscopy*, 87:55–66, 2001.

[30] A. Sanchez-Sevilla, J. Thimonier, M. Marilley, J. Rocca-Serra, and J. Barbet. Accuracy of AFM measurements of the contour length of DNA fragments adsorbed on mica in air and in aqueous buffer. *Ultramicroscopy*, 92:151–158, 2002.

[31] A. Sen and M. Srivastava. *Regression Analysis: Theory, Methods, and Applications*, chapter 12, pages 253–262. Springer, 1990.

[32] A.W.M. Smeulders, L. Dorst, and M. Worring. Measurement and characterisation in vision geometry. In *SPIE Proceedings Series*, volume 3168, 1997.

[33] T.S. Spisz, N. D'Costa, C.K. Seymour, J.H. Hoh, R. Reeves, and I.N. Bankman. Length determination of DNA fragments in atomic force microscope images. In *Proc. Intl. Conf. Image Proc.*, 1997.

[34] T.S. Spisz, Y. Fang, R.H. Reeves, C.K. Seymour, I.N. Bankman, and J.H. Hoh. Automated sizing of DNA fragments in atomic force microscope images. *Med. Biol. Eng. Comput.*, 36:667–672, 1998.

[35] C. Stein. Estimation of the parameters of a multivariate normal distribution: I. estimation of the means. *Ann. Stats.*, 9:1135–1151, 1981.

[36] N. Taleb. *Fooled By Randomness*. Random House, second updated edition, 2005.

[37] D. Tranchida, S. Piccarolo, and R.A.C Deblieck. Some experimental issues of AFM tip blind estimation: the effect of noise and resolution. *Meas. Sci. Technol.*, 17:2630–2636, 2006.

[38] J.S. Villarrubia. Morphological estimation of tip geometry for scanned probe microscopy. *Surf. Sci.*, 321:287–300, 1994.

[39] J.S. Villarrubia. Algorithms for scanned probe microscope image simulation, surface reconstruction, and tip estimation. *J. Res. Nati. Inst. Stand. Technol.*, 102:4:425–454, 1997.

[40] Ch. Wong, P.E. West, K.S. Olson, M.L. Mecartney, and N. Starostina. Tip dilation and AFM capabilities in the characterization of nanoparticles. *JOM*, pages 12–16, 2007.

[41] J.T. Woodward and D.K. Schwartz. Removing drift from scanning probe microscope images of periodic samples. *J. Vac. Sci. Technol. B*, 16:1:51–53, 1998.

[42] M. Worring and A.W.M. Smeuldrers. Digitized circular arcs: Characterization and parameter estimation. *IEEE Trans. Patt. Anal. Mach. Intel.*, 17:6:587–598, 1995.

[43] E. Young. *Seven Blind Mice*. Philomel, 1992.

[44] Z. Zhan, Y. Yang, W.J. Li, Z. Dong, Y. Qu, Y. Wang, and L. Zhou. AFM operating-drift detection and analyses based on automated sequential image processing. Author contact: wen@mae.cuhk.edu.hk, 2006.

[45] G. Zuccheri, A. Scipioni, V. Cavaliere, G. Gargiulo, P. De Santis, and B. Samori. Mapping the intrinsic curvature and flexibility along the DNA chain. *PNAS*, 98:6:3074–3079, 2001.