Image Analysis and Length Estimation of Biomolecules Using AFM

Andrew Sundstrom, *Member, IEEE,* Silvio Cirrone, Salvatore Paxia, Carlin Hsueh, Rachel Kjolby, James K. Gimzewski, Jason Reed, Bud Mishra, *Fellow, IEEE.*

Abstract

There are many examples of problems in pattern analysis for which it is often possible to obtain systematic characterizations, if in addition a small number of useful features or parameters of the image are known a priori or can be estimated reasonably well. Often the relevant features of a particular pattern analysis problem are easy to enumerate, as when statistical structures of the patterns are well understood from the knowledge of the domain. We study a problem from molecular image analysis, where such a domain-dependent understanding may be lacking to some degree and the features must be inferred via machine-learning techniques. In this paper, we propose a rigorous, fully-automated technique for this problem. We are motivated by an application of atomic force microscopy (AFM) image processing needed to solve a central problem in molecular biology, aimed at obtaining the complete transcription profile of a single cell, a snapshot that shows which genes are being expressed and to what degree. Reed *et al* ("Single molecule transcription profiling with AFM.", *Nanotechnology*, 18:4, 2007) showed the transcription profiling problem reduces to making high-precision measurements of biomolecule backbone lengths, correct to within 20-25 bp (6-7.5 nm). Here we present an image processing and length estimation pipeline using AFM that comes close to achieving these measurement tolerances. In particular, we develop a biased length estimator on trained coefficients of a simple linear regression model, biweighted by a Beaton-Tukey function, whose feature universe is constrained by James-Stein shrinkage to avoid overfitting. In terms of extensibility and addressing the model selection problem, this formulation subsumes the models we studied.

Index Terms

AFM, atomic force microscopy, single-molecule, biomolecule, DNA, cDNA, RNA, image processing, digital contour, linear regression, length estimation, biased estimation, Beaton-Tukey, biweight, machine learning, supervised learning

1 Introduction

There are many examples of problems in pattern analysis for which it is often possible to obtain systematic characterizations, if in addition a small number of useful features or parameters of the image are known a priori or can be estimated reasonably well. Examples of such feature-based analysis of patterns occur in human speech [1], genomic data analysis [36], face recognition [27], etc. Often the relevant features of a particular pattern analysis problem are easy to enumerate, as when statistical structures of the patterns are well understood from the knowledge of the domain. We study a problem from molecular image analysis, where such a domain-dependent understanding may be lacking to some degree and the features must be inferred via machine-learning techniques. Similar techniques are beginning to appear in natural image processing [20], [22], neural connectomics analysis [23], population genomics [32], etc., but have not been explored in the area of molecular image analysis, which poses very specific problems of its own. In this paper, we propose a rigorous, fully-automated technique for this problem. In particular, we address several computational questions related to the problem: namely, how can one use standard image processing approaches to get an initial estimate of the length of a dsDNA from its atomic force microscopy (AFM) image and characterize the residual errors? how can one discover a parsimonious set of features that can

- *A. Sundstrom, S. Paxia, and B. Mishra are with the Courant Institute of Mathematical Sciences, NYU, 251 Mercer Street, New York, NY 10012, USA. Email: andrew.sundstrom@cims.nyu.edu, {paxia, mishra}@cs.nyu.edu.*
- *C. Hsueh and J.K. Gimzewski are with the Department of Chemistry and Biochemistry, UCLA, 607 Charles Young Drive East, Los Angeles, CA 90095, USA. Email: hsueh@chem.ucla.edu.*
- *R. Kjolby, J.K. Gimzewski, and J. Reed are with the California NanoSystems Institute (CNSI), 570 Westwood Plaza, Los Angeles, CA 90095, USA. Email: rakjolby@gmail.com, {gim, jreed}@cnsi.ucla.edu.*
- *S. Cirrone is with the Universitá degli Studi di Catania Facoltá di Ingegneria, Catania, Sicily. Email: silvio.cirrone@gmail.com.*

Manuscript received XXX XX, 2011; revised XXX XX, 2011.

explain the residue and improve the length estimate? how can one automatically learn the contributions from a well-chosen subset of features using a training set of calibrating molecules, which may be assumed to contain a large number of "good" examples but possibly corrupted with a few false positives?

We are motivated by an application of image processing needed to solve a central problem in molecular biology, aimed at obtaining 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 $\sim$300,000 mRNA molecules into $\sim$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 AFM to perform single-molecule analysis. Reed *et al* [34] developed such an analysis that classifies each mRNA by the following three steps: (1) synthesize a complementary DNA (cDNA) copy of each mature mRNA, (2) multiply cleave the cDNAs with a restriction enzyme, (3) construct each cDNA 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).

Thus, the solution of the image-processing algorithm needs to be particularly accurate, significantly more than has been demonstrated with previous approaches, and must do so over a wider range of DNA sizes. The approach must be fully-automated, and yet be competitive against the manual or semi-manual approaches that currently outperform computers. The yield from the automatic analysis must be close to perfect; otherwise, the low-copy-number gene expressions will be miscounted. Finally, it has to be compatible with the chemistry and the sensing physics; in other words, the molecules need to be elongated on a sticky uneven surface, may not be fully stretched, may entangle with other molecules, etc. Similarly, AFM may generate multidimensional information (e.g., a magnitude and a phase), may use a wide-variety of scanning strategies, may use parallel scanning with an array of probes, may operate in real time to accommodate low latency and high throughput, etc. None of the previous work we discuss below addresses these issues.

1.1 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 [31] that involve human interaction with image processing and object segmentation algorithms; and automated methods [40], [41], [12], [35], [37], [15], [17], [16], [14] 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 [29]. Some of the automated methods [40], [41], [15], [17], [16], [14] 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. In his masters thesis (2007), Silvio Cirrone innovated 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 edge 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, or molecule backbone in this application.

For nearly 50 years, since Freeman's pioneering work in the image analysis of chain-encoded planar curves [21], 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], [47], [30], [39], [28], [19], [7], [25]. All of the automated processing methods mentioned above employ a

1. Using a tool like NIH Image (http://rsbweb.nih.gov/nih-image/), for example.

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 [40], [41], [12], to those that use one of a variety of parametric estimators [35], [15], [17], [16], [14], to one that takes a signal processing approach based on fast-Fourier transformation followed by Gaussian filtering and normalization [37].

A related focus of investigation involves estimating the *intrinsic curvature* of DNA from AFM images [49], [18]. 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 [47], [39], [25].

1.2 Our approach

We first process the AFM images in a manner typical to the literature: filter the image to extract binary features from background, erode the binary features into 1-D backbone trees, and then prune the trees to extract the backbones. For this last step, we employ the graph-based method used by Cirrone, specified above. The sum of the straight line segments in this backbone give its first length estimate, L_{LS}. Then 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 second backbone length estimate, L_{CS}.

We correct L_{CS} by a linear combination of five features, given below. 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.

Our system implements a meta-approach to the problem of feature-based length estimation. 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], [39] or multigrid convergence [28], [7]. In this way, our approach subsumes those length estimation formulations comprised in small, fixed sets of backbone chain code parameters cited above.

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, aside from computational complexity considerations, there should be no bound on the number of features one applies to the problem.

Our motivation for using the simple machine learning approach of linear regression is manifold:

- It is easy to implement: off-the-shelf libraries are robust, optimized, and have undergone rigorous testing and debugging.
- It is easy to interpret: coefficients are comparatively meaningful as feature weights.
- It is easy to extend: it can support an arbitrary number of image features.
- The Gauss-Markov Theorem guarantees that among all "linear" unbiased estimators, ordinary least squares (OLS) estimates have the smallest variance, and thus, OLS is a best linear unbiased estimator (BLUE).
- The mathematical form of linear regression ($N\vec{a} = \vec{l}$) naturally admits two refinements, aimed at reducing systematic and modeling error, respectively:
  - Empirical Beaton-Tukey biweighting, to address statistical significance: each weight acts on the corresponding *row* of N, the $q \times k$ feature matrix (q calibration molecules by k image features).
  - James-Stein shrinkage, to address overfitting by reducing feature dimensionality: shrinkage uses the mean of each *column* of N to derive a shrinkage factor that acts on the corresponding feature coefficient in $\vec{a}$; features that are noisy (arising from systematic error) or dependent (arising from modeling error) are thus eliminated.

In sum, the training process is supervised learning that is based on a set of examples and counter-examples and the universe of features. Since our method is entirely automated, it lends itself to high-throughput applications.

2 METHODS

Our application, called *AFM Explorer*, uses 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

2. http://www.wxwidgets.org/
3. http://opencvlibrary.sourceforge.net/

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

We outline the steps of *AFM Explorer* below. The image processing 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. (See Figure 1a.) 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`). (See Figure 1b.) To minimize the number of short, noisy fragments (those < 50 nm), we tried many combinations of pixel window dimensions for smoothing and thresholding, eventually choosing the ones given above, which resulted in the best test images.

2.1.2 Erode

To obtain a one-dimensional representation of the molecular backbone contours, we employ the erosion algorithm given in [13], [3], 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. (See Figure 1c.)

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 traversal 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. 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. (See Figure 1d.)

2.1.4 Remove

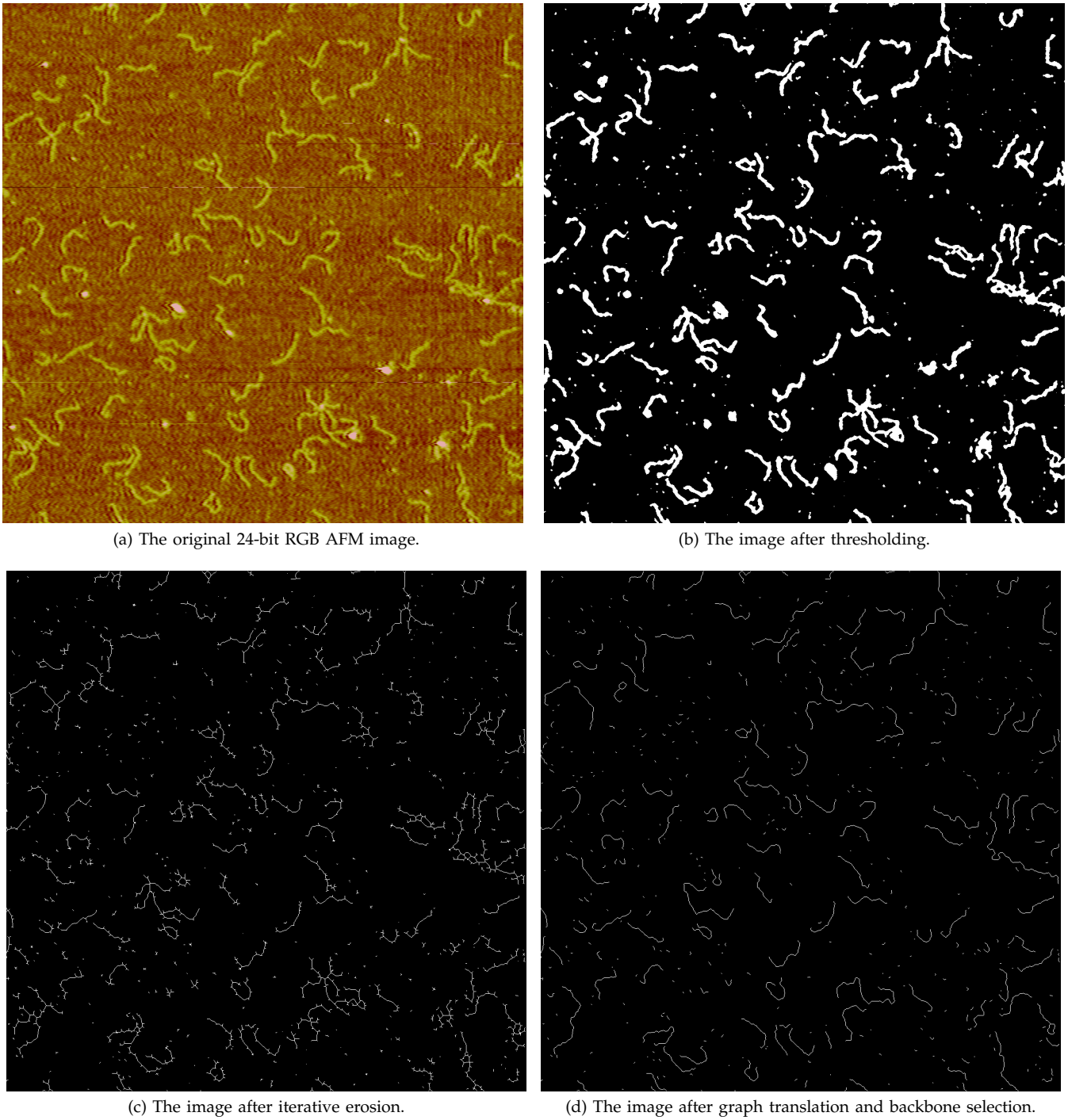
Backbones that stray within 30 pixels from the image boundary are removed, since these represent molecules at the edge of the viewing area that will likely introduce truncated fragments.

2.2 AFM Explorer length estimation pipeline

AFM Explorer uses the length estimation pipeline, whose steps 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-midpoint-to-pixel-midpoint straight line segments, $L_{LS}(\vec{b})$, where horizontal and vertical segments have unit length, and diagonal segments have length $\sqrt{2}$. We then admit a subset $\mathcal{B}' \subset \mathcal{B}$ of backbone pixel vectors, where each $\vec{b}' \in \mathcal{B}'$ meets two criteria: (1) its length is between min and max, set to some mode-dependent values, described below; and (2) it does not intersect with another backbone, according to a simple length heuristic.



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

(b) The image after thresholding.

(c) The image after iterative erosion.

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

Fig. 1: Results of the *AFM Explorer* image processing pipeline.

2.2.2 Secondary estimation using fitted cubic splines

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 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, L_{CS}. Our cubic spline fitting method seems to us a natural instance of the n-point moving polynomial fitting method given in Rivetti, *at al.* [35].

The pipeline has four phases: train, weight, 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 polydisperse 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 polydisperse cDNAs having theoretical lengths in {74.9, 139.6, 223.0, 351.8, 453.1, 583.8} nm.

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

1) the number of horizontal pixel pairs, n_{horz} in $\vec{b'}$
2) the number of vertical pixel pairs, n_{vert} in $\vec{b'}$
3) the number of diagonal pixel pairs, n_{diag} in $\vec{b'}$
4) the number of pixel triples arranged as perpendiculars (i.e., the four orientations of the L shape), n_{perp} in $\vec{b'}$
5) the coefficient of variation for height ($n_{htcv} = \frac{n_{htsd}}{n_{htav}}$ of $\vec{b'}$) is the standard deviation of height divided by the average height, where these are measured as follows: the backbone (1-D) is contained within the binary blob (2-D) that represents the molecule; for each pixel in the backbone (the center), measure its intensity value; upon completing this for all pixels in the backbone, take the arithmetic mean and standard deviation of the measurements.
6) the coefficient of variation for thickness ($n_{tkcv} = \frac{n_{tksd}}{n_{tkav}}$ of $\vec{b'}$) is the standard deviation of thickness divided by the average thickness, where these are measured as follows: the backbone (1-D) is contained within the binary blob (2-D) that represents the molecule; for each pixel in the backbone (the center), extend rays outward in the eight cardinal directions until you reach the boundary of the blob; now consider the sums of the lengths of the four pairs of opposite cardinal direction rays; take the minimum of these four measurements and assign it to the center pixel; upon completing this for all pixels in the backbone, take the arithmetic mean and standard deviation of the measurements assigned to each pixel.

Features 1-3 seem to us natural choices for estimating Euclidean distance, as does Feature 4, especially in light of the discussion of the corner chain estimator in Rivetti, *at al.* [35]. Features 5 and 6 are our estimators of molecular height and thickness, as measured along the extracted backbone; we believe it captures information related to the degree of molecular adsorption onto the mica substrate, and the degree of molecular curvature; it could, in principle, be used to detect overlapping fragments and the binding of markers to the molecule: non-overlapping and unbound fragments would, in principle, have markedly lower average height and thickness.

We train 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 these 6 features: $\{n_{horz}, n_{vert}, n_{diag}, n_{perp}, n_{htcv}, n_{tkcv}\}$, giving $N\vec{a} = \vec{l}$, 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 $\vec{a} = (N^T N)^{-1} N^T \vec{l}$.

This formulation of the estimator, $\mathcal{L'}_T$, assumes all fragments, i.e. their associated feature values, have equal weight, owing to their equivalent validity as observations. However, such an assumption may be challenged on the grounds that upon taking into consideration the difference between the empirically measured null distribution and the actual shape of the distribution in L_{CS} measurements, certain observations appear to be false positives, and others false negatives — a notion formally addressed by robust regression, namely, the Beaton-Tukey formulation.

2.2.4 Weight

Normally, false positive examples appear as ones that deviate significantly from the null-distribution, and if not discarded, can affect the statistical estimators adversely. However, instead of discarding such outliers using sharp-thresholds, and using the filtered examples in the estimator, one may assign to each data point a positive weight that signifies how likely it is that a particular example is an outlier. Such a weighting scheme could be based on the ideas underlying robust M-estimators — a class of central tendency measures that make them resistant to local misbehavior caused by outliers (e.g., false positives). We adapted the Beaton-Tukey biweight [2] — an iteratively reweighted measure — for this purpose of central tendency. We note that other schemes, such as Huber's M-estimator, could have been used with similar performance. Both the biweight and the Huber weight functions are available in standard statistical packages. Here we use Matlab's *robustfit* command with default parameters (weight function "bisquare", using a tuning constant of 4.685).

M-estimator Θ uses these weights to compute the weighted average of sample points: $\Theta = \sum w_i \cdot x_i / \sum w_i$, $0 \leq w_i \leq 1$; the weights are determined in terms a parameter descriptor $u_i = (x_i - \Theta)/\delta$, as follows: δ = MAD (median absolute deviation) and

$$w_i = \begin{cases} [1 - u^2/4.685]^2, & \text{if } |u| \leq 4.685; \\ 0, & \text{otherwise.} \end{cases}$$

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 Shrink

In 1961, James and Stein published their seminal paper [24] 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.

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

$$\delta(\vec{a}) = [\delta_1(\vec{a}), ..., \delta_k(\vec{a})] \tag{1}$$

under the sum of squares error loss

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

In terms of expected loss,

$$R(\vec{\mu}, \delta) = E_{\mu}[L(\vec{\mu}, \delta(\vec{a}))], \tag{3}$$

James and Stein show that when $k \geq 3$, an improved estimator is obtained by a symmetric (or spherical) shrinkage in $\vec{a}$ given by

$$\delta(\vec{a}) = \left[1 - \frac{\kappa(q-k)s^2}{\sum\limits_{i=1}^{q} (N\vec{a})_i^2}\right]^{+} \vec{a}, \tag{4}$$

where

$$\kappa = \frac{(k-2)}{(q-k+2)}, \tag{5}$$

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

$$s^2 = \frac{1}{(q-k)} \sum_{i=1}^{q} (\mathcal{L} - L_{CS_i} - (N\vec{a})_i)^2. \tag{6}$$

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

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 spherical shrinkage is that its performance is guaranteed only in a small subspace of parameter space, 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 proposed a coordinate-based (or truncated) shrinkage method, given by

$$\delta_i^{(f)}(\vec{a}) = \left[1 - \frac{(f-2)s^2 min\{1, \frac{z_{(f)}}{|a_i|}\}}{\sum\limits_{j=1}^{q} (N\vec{m})_j^2}\right]^{+} a_i, \tag{7}$$

where f is a "large fraction" 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

$$s^2 = \frac{1}{(q-k)} \sum_{i=1}^{q} (\mathcal{L} - L_{CS_i} - (N\vec{a})_i)^2, \tag{8}$$

and $\vec{m}_i = min\{a_i, z_{(f)}\}, i = 1, ..., k$. Stein 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.

When we applied spherical and truncated James-Stein shrinkage to our feature coefficients, it did little to reduce the feature dimensionality (i.e., all shrinkage factors were very close to 1). For a summary of these shrinkage factors, see Table 1. From this we inferred our five features had little noise or dependence. Hence, we were confident our linear regression model did not overfit.

TABLE 1: Shrinkage factors and resulting feature correction coefficients for the Linear 6-feature model. There are six pairs of rows: the first row in the pair gives the James-Stein shrinkage factors, and the second row gives the shrinkage factors multiplied by their respective correction coefficients. The first row pair reports the unshrunken correction coefficients, and is given for comparison. Each remaining row pair denotes the result of a shrinkage trial: spherical shrinkage, then four truncated shrinkages (where f is taken from its maximum value, 6, down to its minimum value, 3, as specified by the definition given by James and Stein). The i^{th} column corresponds to the i^{th} correction coefficient.

i	1	2	3	4	5	6
train	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
a_i	-0.13336	-0.0010019	-0.045653	-0.86465	-679.23	38.955
spherical	0.98661	0.98661	0.98661	0.98661	0.98661	0.98661
$\delta_i(\vec{a})$	-0.13158	-0.0009885	-0.045042	-0.85308	-670.14	38.433
truncated $(\mathbf{f = 6})$	0.98660	0.98660	0.98660	0.98660	0.98660	0.98660
$\delta_i^{(6)}(\vec{a})$	-0.13158	-0.00098849	-0.045041	-0.85306	-670.13	38.433
truncated $(\mathbf{f = 5})$	0.98995	0.98995	0.98995	0.98995	0.99942	0.98995
$\delta_i^{(5)}(\vec{a})$	-0.13202	-0.00099184	-0.045194	-0.85596	-678.84	38.563
truncated $(\mathbf{f = 4})$	0.99870	0.99870	0.99870	0.99870	1.00000	0.99997
$\delta_i^{(4)}(\vec{a})$	-0.13319	-0.0010006	-0.045594	-0.86353	-679.23	38.954
truncated $(\mathbf{f = 3})$	0.99937	0.99937	0.99937	0.99990	1.00000	1.00000
$\delta_i^{(3)}(\vec{a})$	-0.13328	-0.0010013	-0.045624	-0.86457	-679.23	38.955

2.2.6 Apply

When the application is in apply mode, the model correction coefficients are locked — they are unadjusted from training — and are loaded from disk. Then each $\vec{b'} \in \mathcal{B'}$ obtains its final estimate, $\mathcal{L'} \in \{\mathcal{L'}_T, \mathcal{L'}_W\}$, from the correction function, $C(\vec{b'}) = a_1 n_{horz}(\vec{b'}) + a_2 n_{vert}(\vec{b'}) + a_3 n_{diag}(\vec{b'}) + a_4 n_{perp}(\vec{b'}) + a_5 n_{htcv}(\vec{b'}) + a_6 n_{tkcv}(\vec{b'})$, and is given by $\mathcal{L'}(\vec{b'}) = L_{CS}(\vec{b'}) + C(\vec{b'})$.

We presently discuss the experimental results of our model's performance, and related factors, on a large set of training and test images.

3 Experiments and 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 2) 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 (e.g. L_{CS}), and then from that, modeling the systematic error intrinsic to the problem.

3.1 Experiments

Our experiments used four data sets, summarized in Table 2. They consist of the following:

1) *Train* data: 17 images comprising a set of 1,865 cDNA fragments having known theoretical lengths {74.9, 139.6, 223.0, 351.8, 453.1, 583.8} nm.

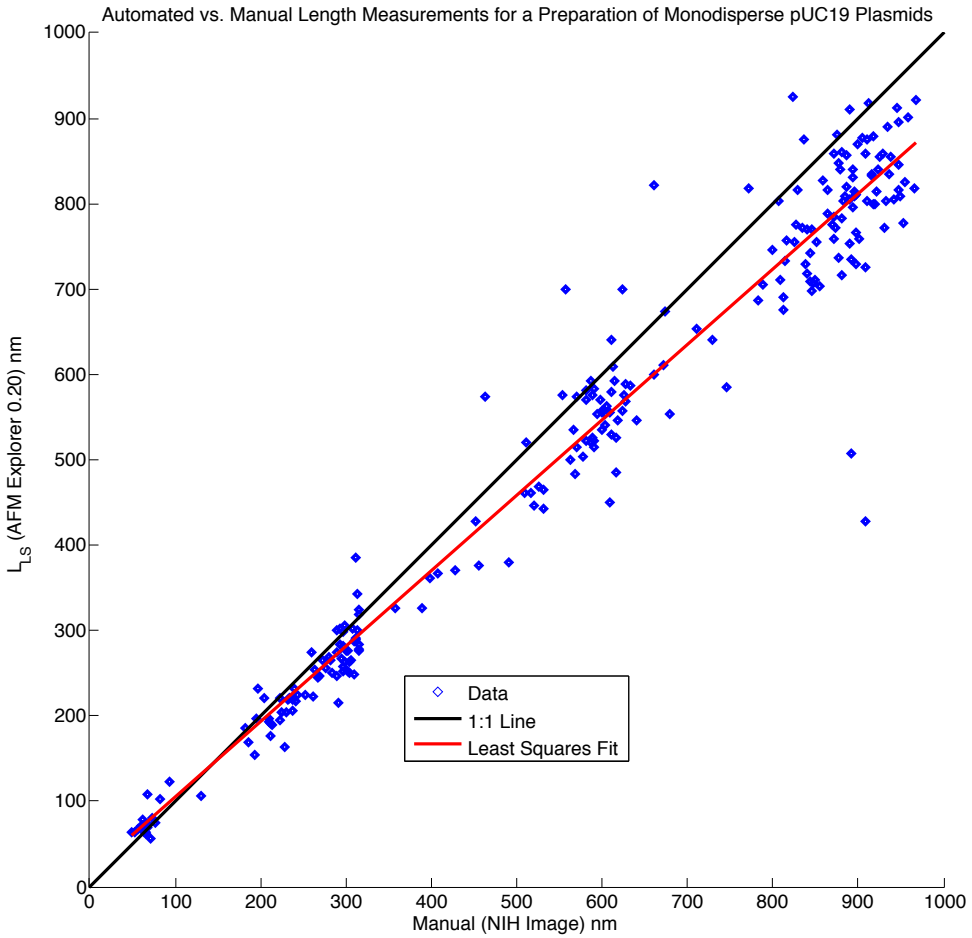


Fig. 2: Early comparative results. Monodisperse pUC19 plasmids were linearized with EcoRi and digested with RsaI restriction enzymes. Fifty AFM images were taken of the resulting fragments, from which 245 fragments were selected and tagged. The lengths given by *AFM Explorer* (version 0.20, producing 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} progressively underestimated fragment backbone length with respect to manual measurements. Note too the proximity of clustering to the theoretically given cleavage points induced by RsaI at 75, 223, and 584 nm; the clustering around 900 nm suggested failed digestion (an intrinsic experimental error).

2) *Test A* data: 20 images comprising a set of 3,415 cDNA fragments having unknown theoretical lengths {33.0, 66.0, 99.0, 132.0, 165.0, 170.6, 198.0, 231.0, 264.0, 297.0, 330.0, 396.0, 500.6} nm.
3) *Test B* data: 9 images comprising a set of 646 cDNA fragments having unknown theoretical lengths {135.3, 258.7, 492.4} nm.
4) *Test C* data: 14 images comprising a set of 1,292 cDNA fragments having unknown theoretical lengths {265.0, 299.0, 444.2, 588.1} nm.

Note that "known" fragment lengths were provided to the length estimation algorithm for training the linear estimator, but the algorithm was blind to "unknown" fragment lengths (known to the experimenter) for testing. Upon acquiring L_{CS} and the 6-feature vector, $\vec{n}$, for each of the 1,865 *Train* backbones, we trained our linear regression model by solving for the six feature correction coefficients, $\vec{a}$. We created a histogram of the cubic spline (L_{CS}) values for the training data (Figure 3).

3.2 Results

The cubic spline (L_{CS}) and estimated length after weighted training ($\mathcal{L}'_W$) results for *Test A*, *Test B*, and *Test C* are summarized in Table 3. In all AFM data, after image processing, there are a large number of short noisy objects. The noise is a combination of electronic and vibration signal noise in the AFM system (very low in our experimental system), and real particles or small bumps on the surface generated by the sample preparation (present in our experimental system) — in general, these are never as long as even the smallest DNA molecules we are interested in measuring.

For each *Test A, B,* and *C*, we created two histograms, corresponding to algorithmic output of L_{CS} and $\mathcal{L}'_W$ (Figures 4, 5, and 6, respectively). We applied a smooth function fit of the histogram data, using Matlab's *ksdensity* function with kernel width 5, to obtain a set of peaks. The locations of these peaks give our estimation of the theoretical fragment lengths in each test. Images were processed using the $0.97\frac{nm}{pix}$ conversion factor.

Measured (L_{CS} and $\mathcal{L}'_W$) versus theoretical lengths for the 15 distinct cDNA fragment lengths in *Tests A, B,* and *C* are shown in Figure 7. Their respective percentage errors ($\frac{|\tau - L_{CS}|}{\tau} \cdot 100$ and $\frac{|\tau - \mathcal{L}'_W|}{\tau} \cdot 100$, given in Table 3) are shown in Figure 8.

We would like to highlight some of our decisions regarding our experiments and error analyses:

1) Test A, $\tau = \{198.0\}$ nm: no peak was detected using our chosen smoothing settings; thus it is a false negative and we did not report this error in Table 3.
2) Test A, $\tau = \{165.0, 170.6\}$ nm: the peak finding detected only one of the two peaks because these were so close together; thus we used their arithmetic mean ($\mu = 167.8$ nm) as the "known" theoretical value for the sake of reporting the corresponding errors in Table 3.
3) Test A, $\tau = \{396.0, 500.6\}$ nm: the abundances of these two species are too low to be meaningful; thus we did not report these errors in Table 3. This is an inherent property of the sample, not our experimental method: Test A is a 100 bp sizing ladder used for size standards in gel electrophoresis; by design the shorter species have higher abundance, not an artifact of sample preparation or data processing.
4) Test C, $\tau = \{444.2\}$ nm: peaks were detected at $L_{CS} = 489.44$ and $\mathcal{L}'_W = 469.95$, giving respective errors of: 45.24 nm and 25.75 nm (10.19% and 5.80%) — obvious outlier errors. Since the original sequence provided for the plasmid by the vendor did not reconcile with our measurements, we decided to investigate further. It turns out that the plasmid we used had a modification that was not documented, thus the detected peaks represented a true unknown. This can happen in cases where the plasmid is obtained from a large collection (as ours was) and the vendor's quality control is not 100% effective. We obtained the sequence of the plasmid ourselves and discovered the correct theoretical length is 475.60 nm instead of 444.20 nm. The corrected theoretical length is reported in Table 2, and the corrected error values are reported in Table 3 and Figures 7 and 8.

TABLE 2: Training and test data sets used in experiments. Each data set's label, number of images, number of admissible fragments, and theoretical lengths of fragments (τ) is given.

Data Set	Images	Fragments	τ (nm)
Train	17	1,865	{74.9, 139.6, 223.0, 351.8, 453.1, 583.8}
Test A	20	3,415	{33.0, 66.0, 99.0, 132.0, 165.0, 170.6, 198.0, 231.0, 264.0, 297.0, 330.0, 396.0, 500.6}
Test B	9	646	{135.3, 258.7, 492.4}
Test C	14	1,292	{265.0, 299.0, 475.6, 588.1}

4 DISCUSSION

In the problem described in this paper, there are two principal sources of error: bias from the method of estimation (the extrinsic factors), and systematic error (the intrinsic factors) that come from chemistry experimental error, and AFM operation and measurement error. We have given a BLUE estimator for molecular backbone contour length, namely the piecewise cubic spline fitting measure, L_{CS}. But this estimator gets us only part way to the answer, since systematic error underlies all such measurements. 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, $\mathcal{L}'_T$. By weighting the linear regression training based on computed Beaton-Tukey biweights, we created another estimator, $\mathcal{L}'_W$, that further improves performance. These estimators were trained on the six features given above. James-Stein shrinkage analysis gave almost undetectable improvement, suggesting the six features were neither noisy nor dependent (Table 1). 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.

4.1 Comparison with Other Studies

In the following discussion, we define: the known *theoretical length* of a given molecular fragment to be τ; the *best reported length estimator* in a given study to be $\mathcal{L}$; the *error in nm* for a given measurement with respect to a given τ to be $|\tau - \mathcal{L}|$; and the *error percentage* for the given measurement with respect to a given τ to be $\frac{|\tau - \mathcal{L}|}{\tau} \cdot 100$.

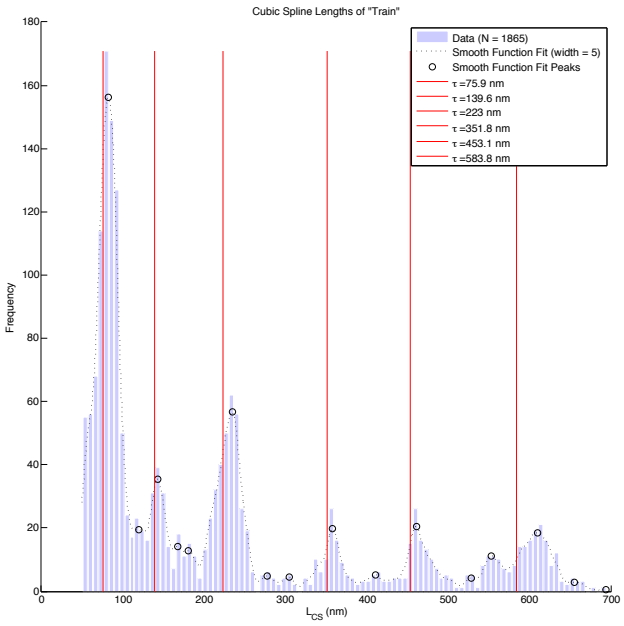
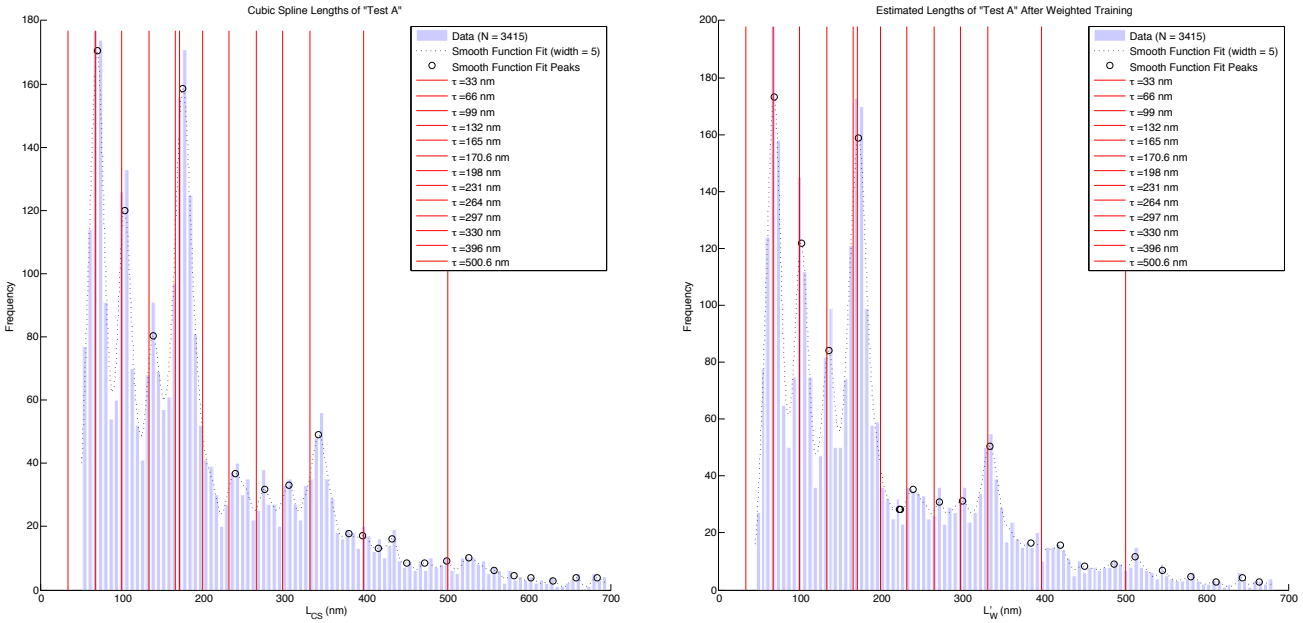


Fig. 3: Length histogram of L_{CS} for *Train*.



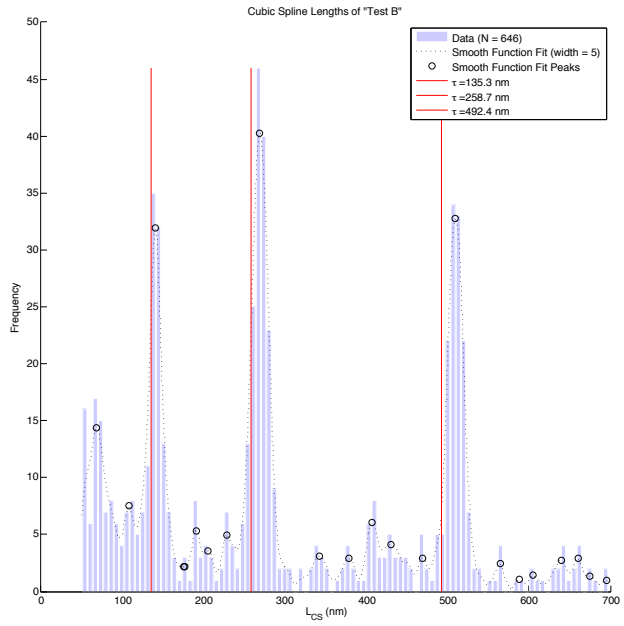
(a) Length histogram of L_{CS} for *Test A*.

(b) Length histogram of $\mathcal{L}'_W$ for *Test A*.

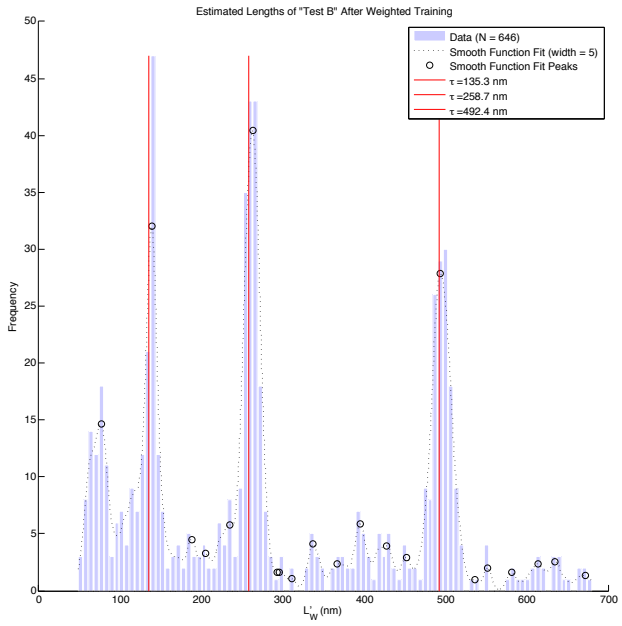
Fig. 4: Estimation of the theoretical fragment lengths in *Test A*.

Among the automated methods studied, Fang *et al* (1998) [12] have published the most comprehensive work on this issue to date, where they achieved an error percentage in the range $[1.67, 10.67]\%$ for *13 distinct* theoretical lengths of fragments in the length range [30.00, 750.00] nm. Sanchez-Sevilla *et al* (2002) [37] reported error percentage in the range $[0.56, 1.46]\%$ for *two distinct* theoretical lengths of fragments in the length range $[206.00, 355.00]$ nm. More impressively, Ficarra *et al* (2005) [14] described a method that achieved better sizing, reporting error percentage in the range $[0.31, 1.18]\%$ for *two distinct* theoretical lengths of fragments in the length range $[633.40, 1098.00]$. We report error percentage in the range $[0.28, 3.24]\%$ for *15 distinct* theoretical lengths of fragments in the length range [66.00, 588.10] nm. We present all comparative results in Table 4 and Figure 9, where for our study we define $\mathcal{L}$ to be $\mathcal{L}'_W$.

We should note the trends that are evident in Figure 9. Viewed as a function of fragment length, error percentage: increases for Fang, *et al* [12] inside a wide dispersion of $N = 16$ data points; decreases for Sanchez-Sevilla, *et al* [37]

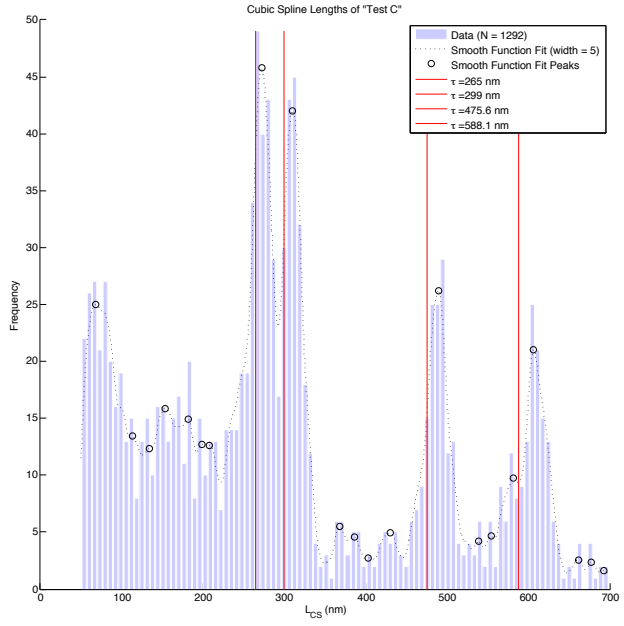


(a) Length histogram of L_{CS} for *Test B*.

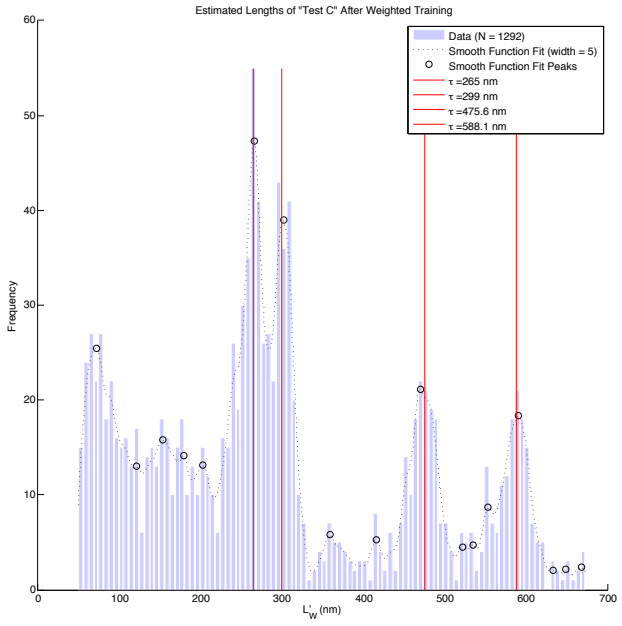


(b) Length histogram of $\mathcal{L}'_W$ for *Test B*.

Fig. 5: Estimation of the theoretical fragment lengths in *Test B*.



(a) Length histogram of L_{CS} for *Test C*.



(b) Length histogram of $\mathcal{L}'_W$ for *Test C*.

Fig. 6: Estimation of the theoretical fragment lengths in *Test C*.

inside a dispersion of $N = 2$ data points; increases for Ficarra, *et al* [14] inside a dispersion of $N = 3$ data points; and decreases for our results inside a narrow dispersion of $N = 15$ data points. Our trend gives us reason to believe that our estimation method would yield accurate (< 1 error percentage) length measurements for molecular fragments larger than 600 nm. While our results do not strictly speaking outperform those reported by Sanchez-Sevilla, *et al* [37] and Ficarra, *et al* [14], we believe our results achieve nearly the same length measurement accuracy through a novel supervisory learning approach that benefits from empirical-Bayesian statistical insights. We should also note that we (and Fang, *et al* [12]) tested our approach more comprehensively than did Sanchez-Sevilla, *et al* [37] and Ficarra, *et al* [14] (i.e., more fragments, wider range of sizes, etc.)

The other studies we found took the image processing aspect of the problem to the limit. The approach taken by Ficarra, *et al* [14] is a good example. These studies also use simple length correction methods to address the errors

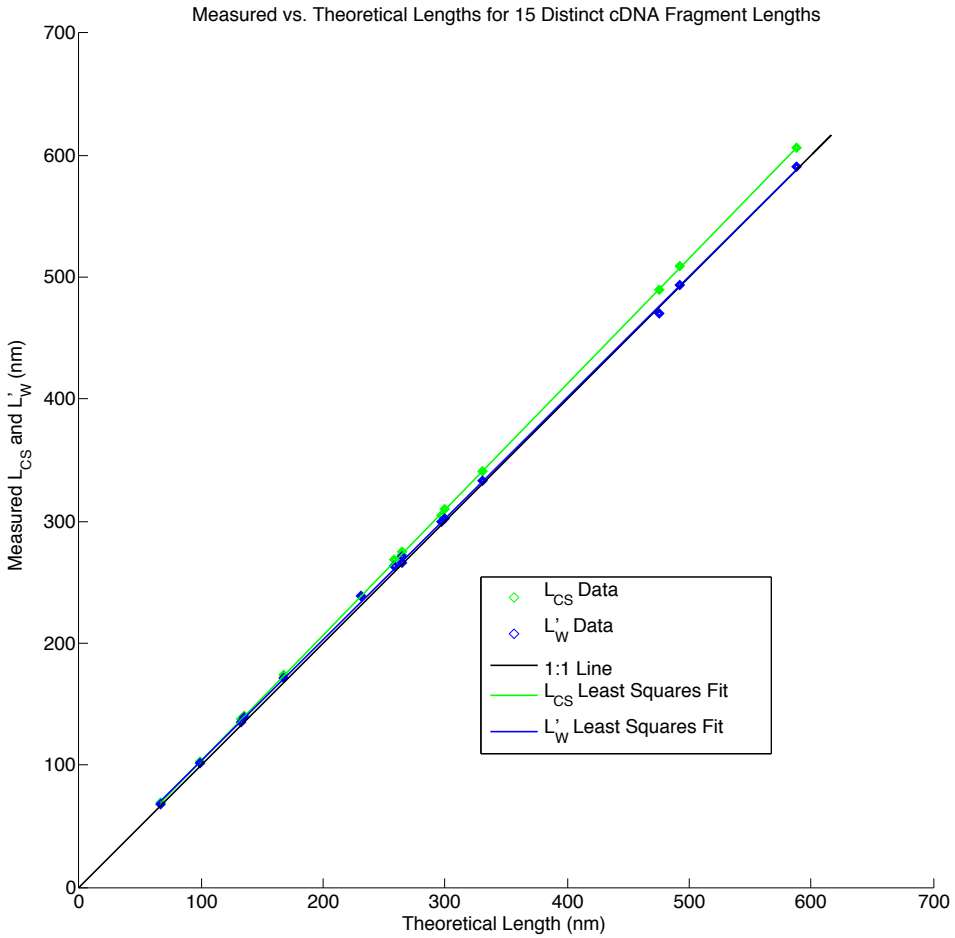


Fig. 7: Measured (L_{CS} and $\mathcal{L}'_W$) versus theoretical lengths for the 15 distinct cDNA fragment lengths in *Tests A, B,* and *C*.

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. This is as far as they go to address the tip convolution problem, discussed below; 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. In our current AFM system, thermal drift is negligible over the time scale for one molecule to be imaged (a few seconds).

4.2 Unique aspects of AFM

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* and *tip-surface adhesion*. Second, none attempt to model tip convolution effects directly and appropriately deconvolve the image, though the problem is widely acknowledged [26], [43], [8], [42], [45] and algorithms designed precisely for this purpose exist [44]. Third, none attempt to model thermal drift directly and perform the appropriate deblurring of the image (locally or globally) though this problem too is widely acknowledged [6], [46], [33], [48] and an assortment of well-suited algorithms for this exist, namely Carasso's SECB algorithm [4], [5]. Fourth, experimenters can use closed-loop scanning settings in their protocols, to reduce the effects of mechanical 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 obtain more accurate length estimates. In addition, there are problems implicit to the chemistry, namely, it is not well understood how a three-dimensional DNA molecule adsorbs onto a substrate

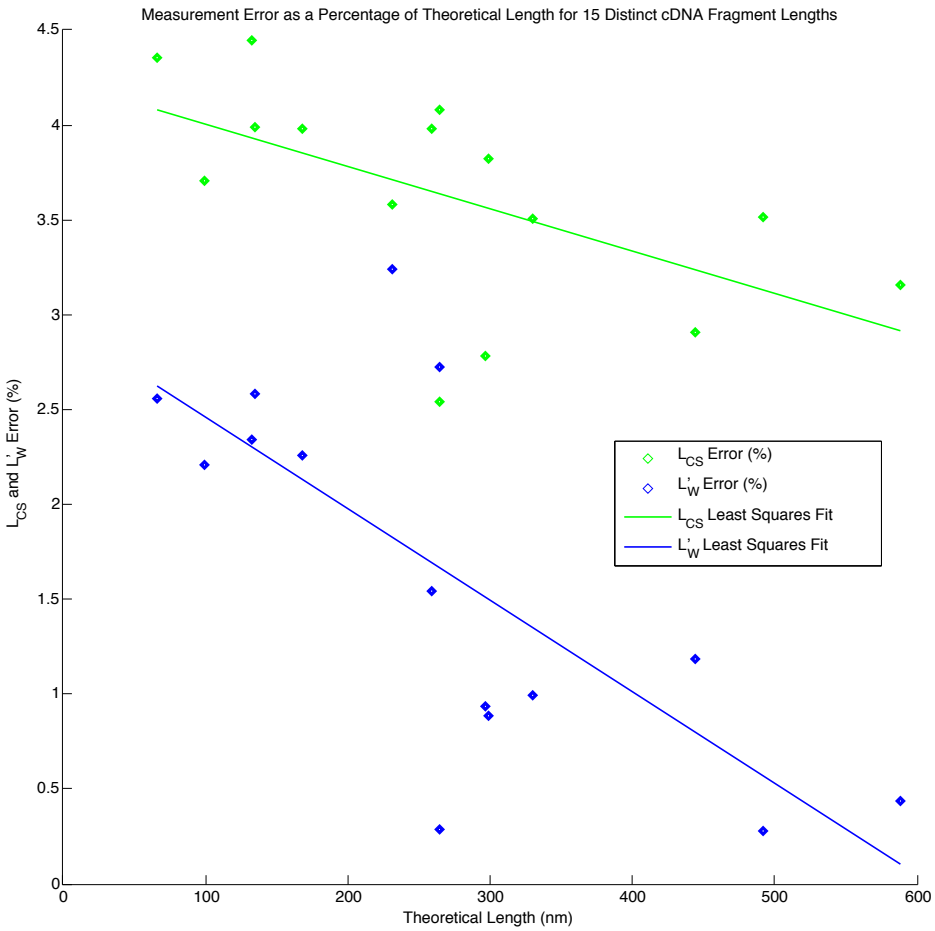


Fig. 8: Errors of measured (L_{CS} and $\mathcal{L}'_W$) lengths for the 15 distinct cDNA fragment lengths in *Tests A*, *B*, and *C*, expressed as a percentage of their respective theoretical fragment lengths.

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.

5 Summary and Conclusion

The approach developed in this paper builds upon the concept of "supervised learning", a widely used methodology in machine learning with applications to systems biology and internet tools. In this methodology, a supervisor trains a machine learning algorithm to select a model by looking for significant features from large corpora of correct examples. In this way, we attempt to learn various subtle features in the data and how these features are related to systematic error; these models are then used to rectify the systematic errors. However, if the supervisor is imperfect, and allows some number of false positive examples, then these outliers can confound the machine learning algorithm, as it attempts to compensate for the presumed systematic errors even when there is no relationship between the perceived errors in these false positive examples and the extracted features. The resulting process would then lead to an undesirable bias in the statistical estimation. The solution to these problems would require either manual marking of the correct examples, or some form of outlier detection and robust estimation process. Our approach involves a weighted scheme, in which a weight is assigned to each training example, and corresponds to the probability that the putative training example belongs to a particular theoretical length. We built an empirical method for assigning weight around the Beaton-Tukey biweighting algorithm. In this scheme, the statistical estimator algorithm was suitably modified to minimize a weighted sum-of-square error. Afterward, James-Stein shrinkage provides a means of constraining the universe of features to retain those that informatively describe molecular backbone length correction.

Acknowledgments

The research reported in this paper was supported by: an NIH-NHGRI and an NSF (CDI Type II) grant to B. Mishra; an NIH grant (GM080999) to J.K. Gimzewski, B. Mishra, and J. Reed; and an NIH grant (R01GM094388) to J. Reed.

TABLE 3: Experimental results. Rows are divided into three groups, corresponding to *Tests A, B,* and *C*, indicated by the first column. Columns are then divided into three groups, corresponding to lengths (cubic spline length L_{CS}, estimated length after weighted training $\mathcal{L}'_W$, and theoretical length τ), errors measured in nm ($|\tau - L_{CS}|$ and $|\tau - \mathcal{L}'_W|$, respectively), and errors measured in percentage of corresponding theoretical fragment length ($\frac{|\tau - L_{CS}|}{\tau} \cdot 100$ and $\frac{|\tau - \mathcal{L}'_W|}{\tau} \cdot 100$, respectively). Results from the "Length (nm)" columns are plotted in Figure 7. Results from the "Error (%)" columns are plotted in Figure 8.

	Length (nm)			Error (nm)		Error (%)	
Test	L_{CS}	$\mathcal{L}'_W$	τ	L_{CS}	$\mathcal{L}'_W$	L_{CS}	$\mathcal{L}'_W$
A	68.87	67.69	66.00	2.87	1.69	4.35	2.56
A	102.67	101.19	99.00	3.67	2.19	3.71	2.21
A	137.87	135.09	132.00	5.87	3.09	4.45	2.34
A	174.47	171.59	167.80	6.67	3.79	3.98	2.26
A	239.27	238.49	231.00	8.27	7.49	3.58	3.24
A	274.77	271.19	264.00	10.77	7.19	4.08	2.72
A	305.27	299.79	297.00	8.27	2.79	2.79	0.94
A	341.57	333.29	330.00	11.57	3.29	3.51	1.00
B	140.70	138.79	135.30	5.40	3.49	3.99	2.58
B	269.00	262.69	258.70	10.30	3.99	3.98	1.54
B	509.70	493.79	492.40	17.30	1.39	3.51	0.28
C	271.74	265.75	265.00	6.74	0.75	2.54	0.28
C	310.44	301.65	299.00	11.44	2.65	3.83	0.89
C	489.44	469.95	475.60	13.84	5.65	2.91	1.19
C	606.64	590.65	588.10	18.54	2.55	3.15	0.43

REFERENCES

[1] B. Atal and L. Rabiner. A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 24:3:201–212, 1976. ISSN:0096-3518.

[2] A.E. Beaton and J.W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16:2:147–185, 1974.

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

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

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

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

[7] 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.

[8] 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.

[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] B. Efron. Large-scale simultaneous hypothesis testing: The choice of a null hypothesis. *J. Amer. Stat. Assoc.*, 99:465:96–104, 2004.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

TABLE 4: Automated DNA sizing accuracy claims. Columns give the authors of the studies, the theoretical DNA fragment lengths under investigation (τ), and the errors of their best estimator ($\mathcal{L}$) obtained in their respective experiments. Errors measured in nm are calculated as $|\tau - \mathcal{L}|$). Errors measured in percentage of corresponding theoretical fragment length are calculated as $\frac{|\tau-\mathcal{L}|}{\tau} \cdot 100$. All calculations assume 3 nm = 10 bp. Results from the "Error (%)" column are plotted in Figure 9.

Author (year)	Fragment Length (nm)	Error (nm)	Error (%)
Fang, *et al* (1998)	30.00	3.00	10.00
	60.00	1.00	1.67
	60.00	6.00	10.00
	75.00	3.00	4.00
	90.00	6.00	6.67
	90.00	7.00	7.78
	120.00	10.00	8.33
	150.00	8.00	5.33
	180.00	7.00	3.89
	210.00	12.00	5.71
	240.00	22.00	9.17
	300.00	20.00	6.67
	300.00	32.00	10.67
	450.00	32.00	7.11
	600.00	57.00	9.50
	750.00	38.00	5.07
Sanchez-Sevilla, *et al* (2002)	206.00	3.00	1.46
	355.00	2.00	0.56
Ficarra, *et al* (2005)	633.40	2.10	0.33
	633.40	2.00	0.31
	1098.00	13.00	1.18
Sundstrom, *et al* (2012)	66.00	1.69	2.56
	99.00	2.19	2.21
	132.00	3.09	2.34
	135.30	3.49	2.58
	167.80	3.79	2.26
	231.00	7.49	3.24
	258.70	3.99	1.54
	264.00	7.19	2.72
	265.00	0.75	0.28
	297.00	2.79	0.94
	299.00	2.65	0.89
	330.00	3.29	1.00
	475.60	5.65	1.19
	492.40	1.39	0.28
	588.10	2.55	0.43

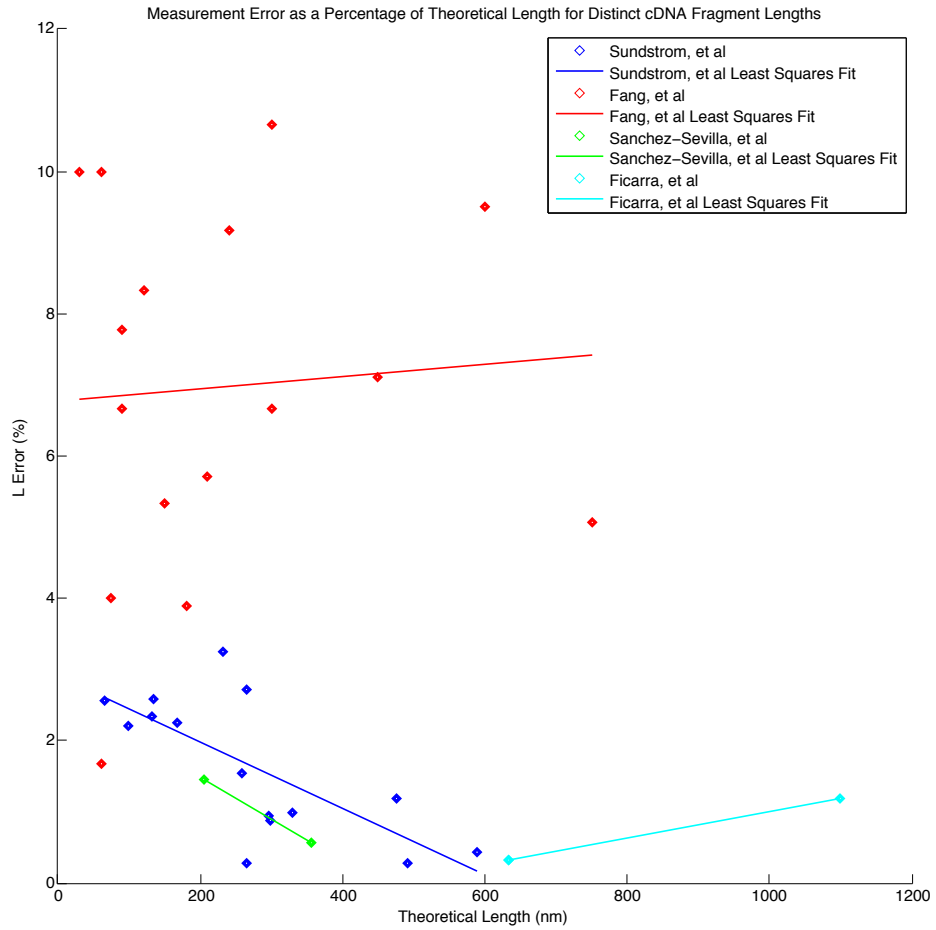


Fig. 9: Errors of measured lengths for distinct cDNA fragment lengths across cited studies, expressed as a percentage of their respective theoretical fragment lengths.

[19] 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.

[20] D. Forsyth, J. Malik, M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, and C. Bregler. *Lecture Notes in Computer Science: Object Representation in Computer Vision II*, chapter Finding Pictures of Objects in Large Collections of Images, pages 335–360. Springer, 1996. doi:10.1007/3-540-61750-7_36.

[21] 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.

[22] P. Hanchuan. Bioimage informatics: a new area of engineering biology. *Bioinformatics*, 24:17:1827–1836, 2008. doi:10.1093/bioinformatics/btn346.

[23] V. Jain, J.F. Murray, F. Roth, S. Turaga, V. Zhigulin, K. Briggman, M. Helmstaedter, W. Denk, and H.S. Seung. Supervised learning of image restoration with convolutional networks. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, 2007. doi:10.1109/ICCV.2007.4408909.

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

[25] 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.

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

[27] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. *PAMI*, 12:1:103–108, 1990.

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

[29] 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.

[30] 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.

[31] 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.

[32] P. Marjoram, J. Molitor, V. Plagnol, and S. Traver'e. Markov chain monte carlo without likelihoods. *PNAS*, 100:26:15324–15328, 2003. doi:10.1073/pnas.0306899100.

[33] 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.

[34] 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.

[35] 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.
[36] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23:19:2507–2517, 2007. doi:10.1093/bioinformatics/btm344.
[37] 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.
[38] A. Sen and M. Srivastava. *Regression Analysis: Theory, Methods, and Applications*, chapter 12, pages 253–262. Springer, 1990.
[39] A.W.M. Smeulders, L. Dorst, and M. Worring. Measurement and characterisation in vision geometry. In *SPIE Proceedings Series*, volume 3168, 1997.
[40] 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.
[41] 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.
[42] 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.
[43] J.S. Villarrubia. Morphological estimation of tip geometry for scanned probe microscopy. *Surf. Sci.*, 321:287–300, 1994.
[44] 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.
[45] 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.
[46] 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.
[47] 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.
[48] 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.
[49] 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.