Supplementary Materials for 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*

1 METHODS

Our application, called *AFM Explorer*, uses the *wxWidgets*[1] and *OpenCV*[2] 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.

1.1 AFM Explorer image processing pipeline

We outline the steps of *AFM Explorer* below. The image processing pipeline has four phases:

1.1.1 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 Supplementary 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 Supplementary 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. In our current implementation, we found that a 5×5 kernel (for smoothing) and a 31×31 kernel (for thresholding) works well with the images of $0.97\frac{nm}{pixel}$ resolution that we use uniformly throughout this study; other kernel sizes might be more appropriate for different resolutions.

1.1.2 Erode

To obtain a one-dimensional representation of the molecular backbone contours, we employ the erosion algorithm given in [7], [2], 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 Supplementary Figure 1c.)

1.1.3 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

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

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 Supplementary Figure 1d.)

1.1.4 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.

1.2 AFM Explorer length estimation pipeline

AFM Explorer uses the length estimation pipeline, whose steps we outline the steps below.

1.2.1 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.

1.2.2 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.* [11].

The pipeline has four phases: train, weight, shrink, and apply.

1.2.3 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. All features were computed after image processing, using the binary image, resulting from thresholding, where molecular backbone objects are white pixels ({R, G, B} = {255, 255, 255}) on a black ({R, G, B} = {0, 0, 0}) background. The coefficient of variation for height (feature number 5 below) also uses the corresponding pixels in the 8-bit grayscale image, resulting from smoothing, to obtain the intensity values that represent height. 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.* [11]. 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 princi-

ple, 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 gives a trained estimator, ${\mathcal{L}'}_T$, as computed using the q training molecules in the Apply phase below.

This formulation of ${\mathcal{L}'}_T$ 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.

1.2.4 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 [1] — 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: $\delta =$ 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 the context of our system, the $x_i, i = 1, ..., q$ are the L_{CS} of the q calibrating molecule backbones in the training set. Each molecule is assigned a weight corresponding to each known theoretical length in the training set. For example, if the training set is comprised of 1,000 molecules from 5 distinct species, then we compute a $1,000 \times 5$ weight matrix. Summing across rows, if any of the rows has sum equal to zero, then the corresponding molecule is discarded from the training set. Of the q training molecules, q' remain. This gives a weighted trained estimator, ${\mathcal{L}'}_W$, as computed using the $q' \leq q$ training molecules in the Apply phase below.

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.

1.2.5 Shrink

In 1961, James and Stein published their seminal paper [8] 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

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

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

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

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

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

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 Supplementary 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.

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

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 [9], [13], [6], [12], [15] and algorithms designed precisely for this purpose exist [14]. 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 [5], [16], [10], [17] and an assortment of well-suited algorithms for this exist, namely Carasso's SECB algorithm [3], [4]. 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 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.

3 Sample Preparation and Imaging

All DNA samples are suspended in Tri buffer (pH 7.6) with 10 mM MgCl2 to permit the molecules to stably adsorb the mica substrate. Samples are deposited in a volume of 100 uL on freshly-cleaved mica substrates and allowed to incubate for 10 minutes without drying, followed by gentle washing (3X) with purified water. The substrates are then dried with a stream of dry nitrogen and soft baked at 120C for 15 minutes to eliminate excess adsorbed water. Samples are imaged in tapping mode in air with the Dimension ICON AFM (Bruker Metrology) using $k = \sim 3$ N/m silicon nitride tips (Nanosensors). Scan sizes were 3x3 microns, imaged at 2 Hz. Image resolution was 2 nm/pixel.

4 Software

Our image processing and length estimation software is written in C++ and Matlab. We will make our code available to interested parties via our website, http://bioinformatics.nyu.edu/.

References

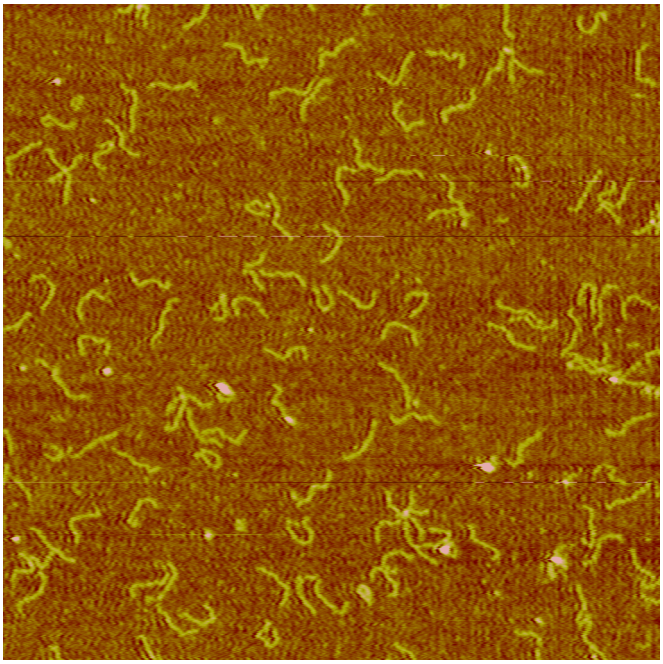
[1] A.E. Beaton and J.W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16:2:147–185, 1974.

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

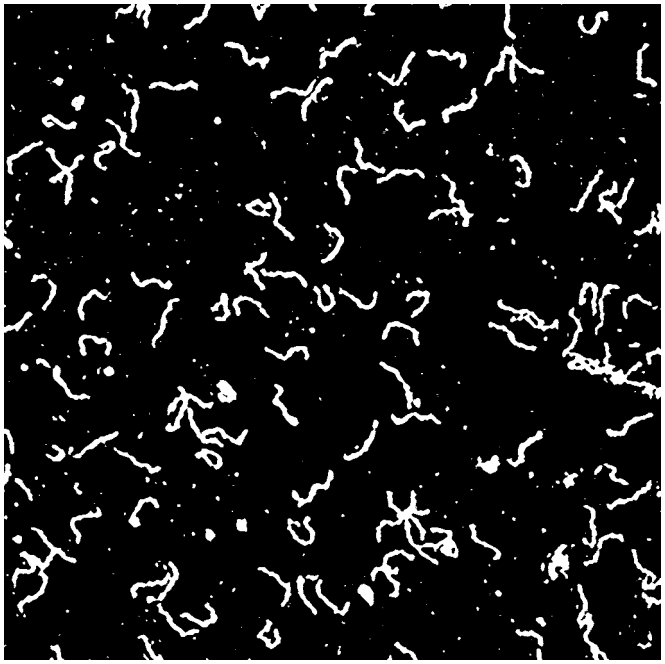
[3] A.S. Carasso. Error bounds in nonsmooth image deblurring. *SIAM J. Math. Anal.*, 28:3:656–668, 1997.

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

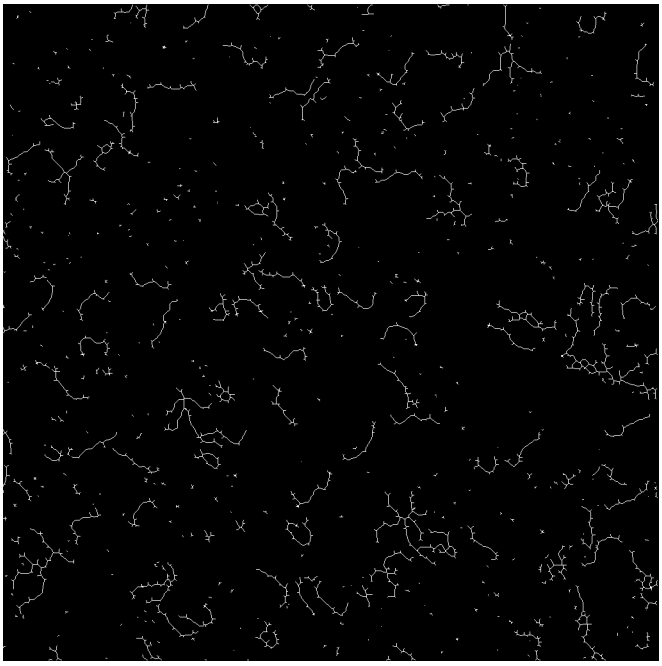
[5] Y. Chen and W. Huang. Application of a novel nonperiodic grating in scanning probe microscopy drift measurement. *Rev. Sci. Instr.*, 78:7, 2007.



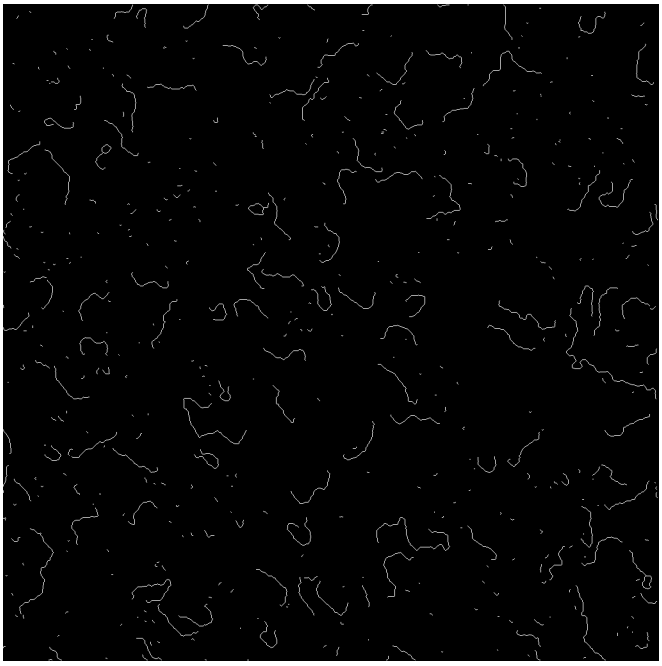
(a) The original 24-bit RGB AFM image.



(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.

[6] 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.

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

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

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

[10] 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.

[11] 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.

[12] 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.

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

[14] J.S. Villarrubia. Algorithms for scanned probe microscope image simulation, surface reconstruction, and tip estimation. *J. Res. Nati.*

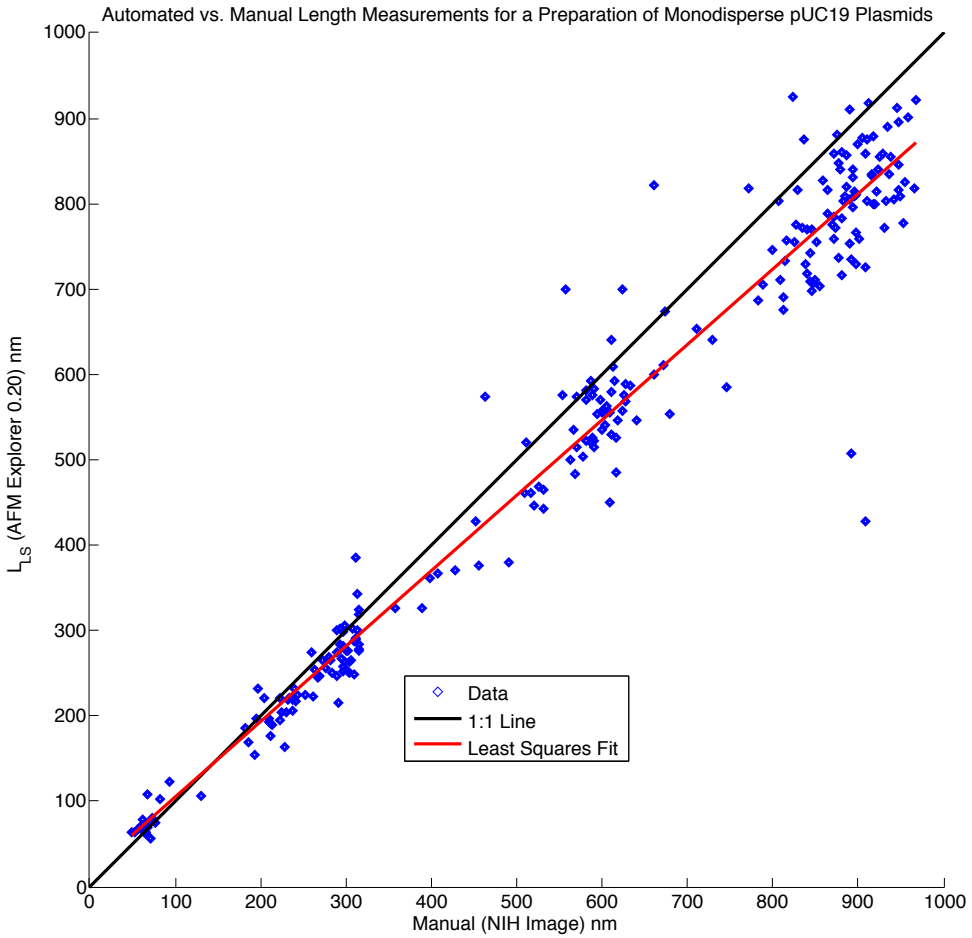


Fig. 2: Early comparative results. Monodisperse pUC19 plasmids were linearized with EcoRi and digested with RsaI restriction enzymes. Fifty AFM images were taken of the resulting fragments, from which 245 fragments were selected and tagged. The lengths given by *AFM Explorer* (version 0.20, producing 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). Note that these results were obtained using a prototype version of our software, and so this figure presents preliminary data using L_{LS}, before the other metrics ($L_{CS}, \mathcal{L}'_T, \mathcal{L}'_W$) were developed. Also note that the set of images used to produce this figure were not available for processing by subsequent versions of our software.

Inst. Stand. Technol., 102:4:425–454, 1997.

[15] 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.

[16] 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.

[17] 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.

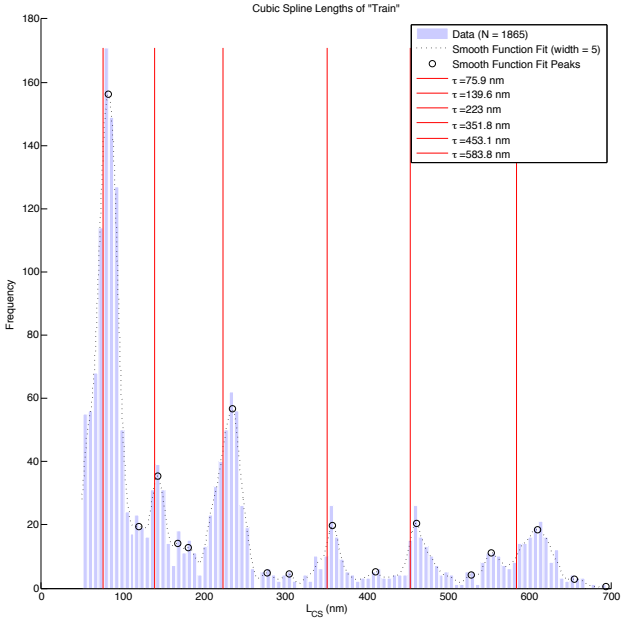
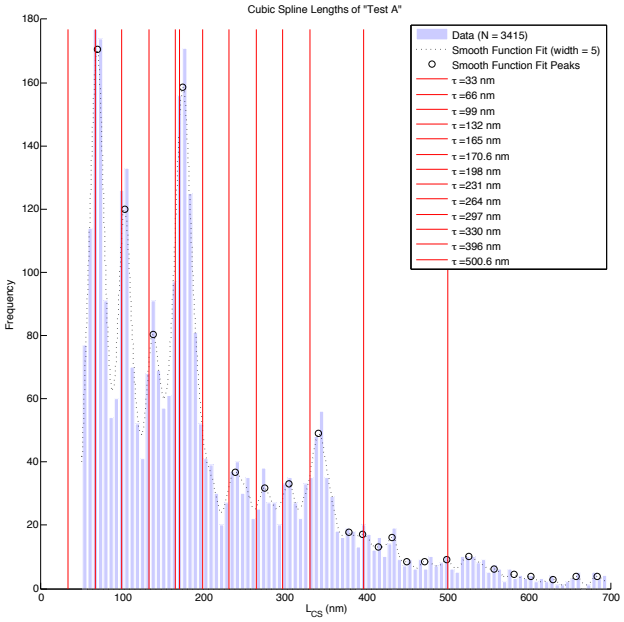
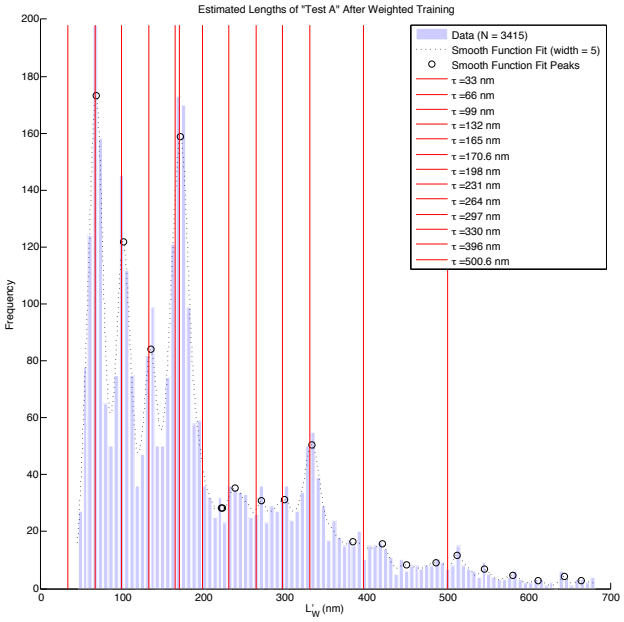


Fig. 3: Length histogram of L_{CS} for *Train*.

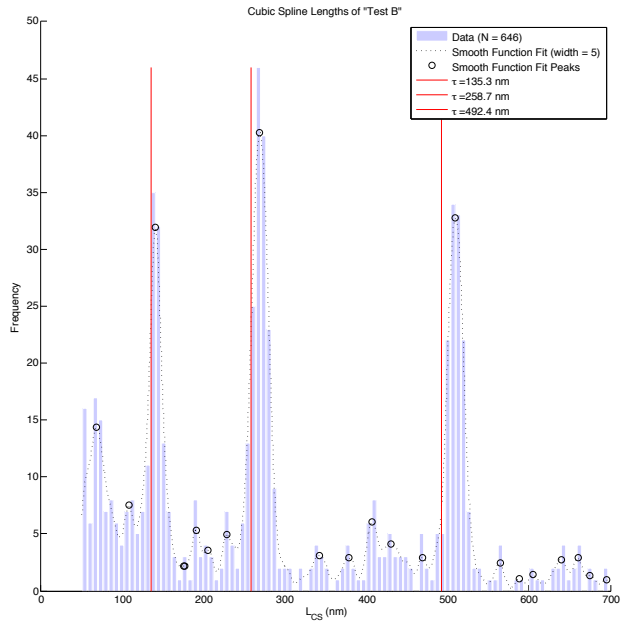


(a) Length histogram of L_{CS} for *Test A*.

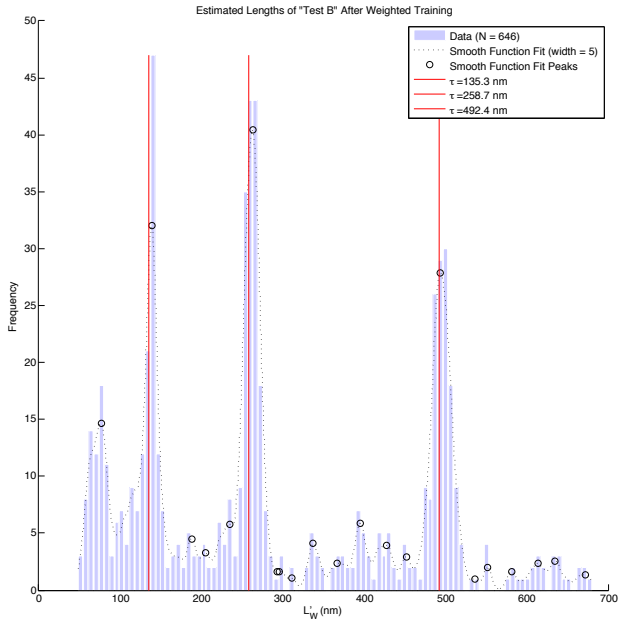


(b) Length histogram of $\mathcal{L}'_W$ for *Test A*.

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

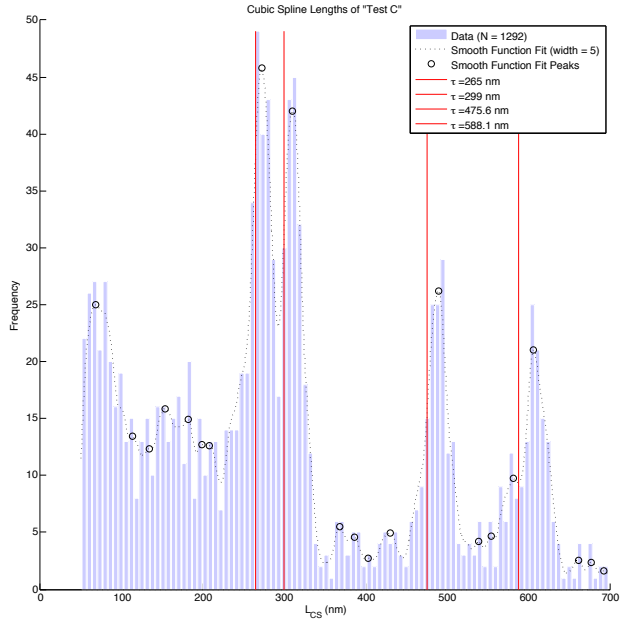


(a) Length histogram of L_{CS} for *Test B*.

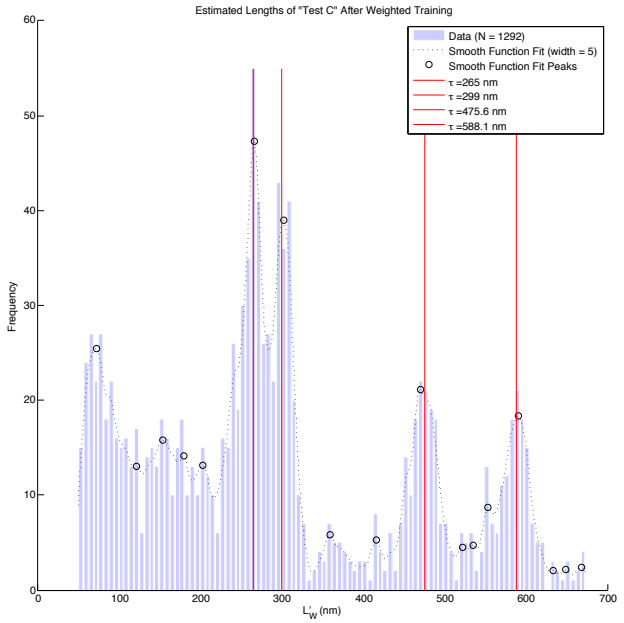


(b) Length histogram of $\mathcal{L}'_W$ for *Test B*.

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*.

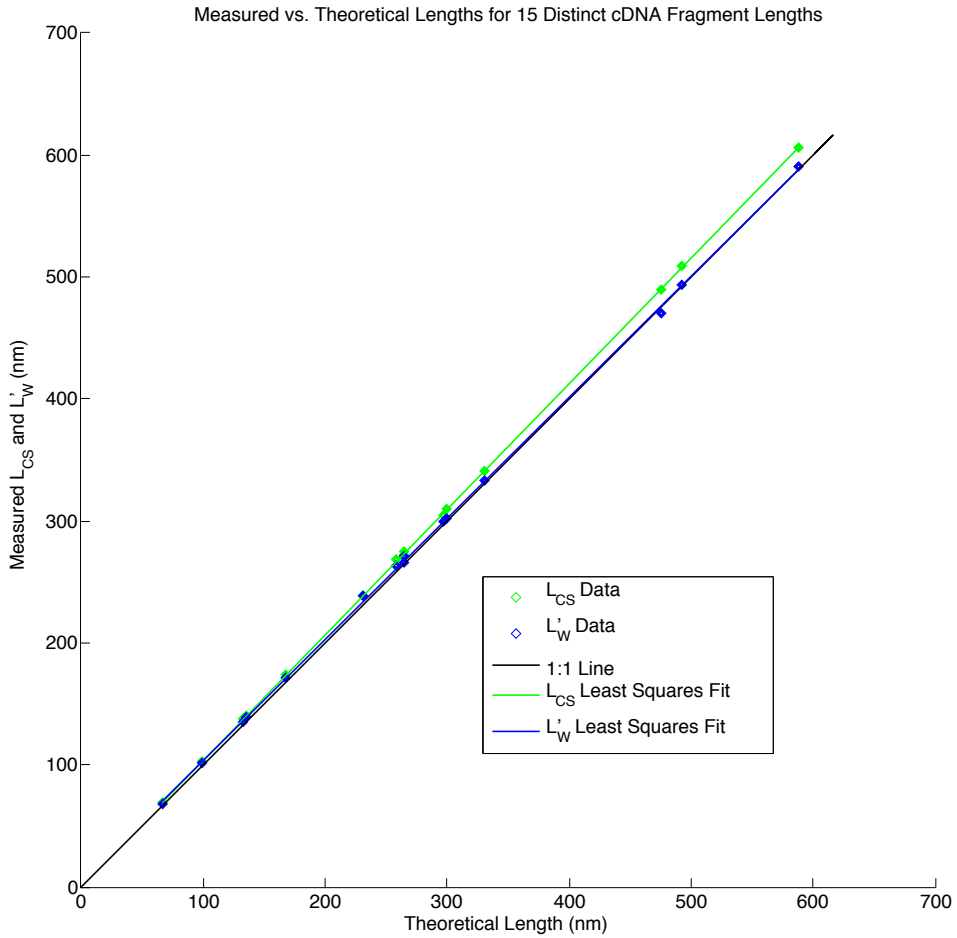


Fig. 7: Measured (L_{CS} and $\mathcal{L}'_W$) versus theoretical lengths for the 15 distinct cDNA fragment lengths in *Tests A*, *B*, and *C*.

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

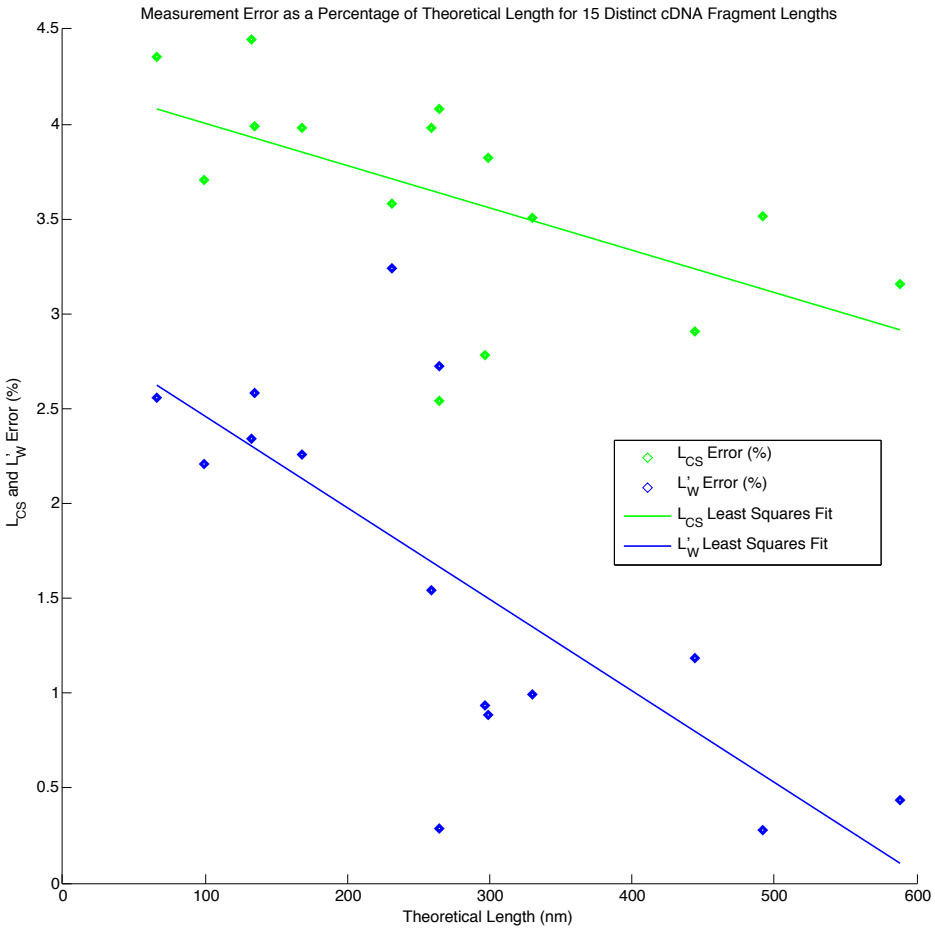


Fig. 8: Errors of measured (L_{CS} and $\mathcal{L}'_W$) lengths for the 15 distinct cDNA fragment lengths in *Tests A*, *B*, and *C*, expressed as a percentage of their respective theoretical fragment lengths.

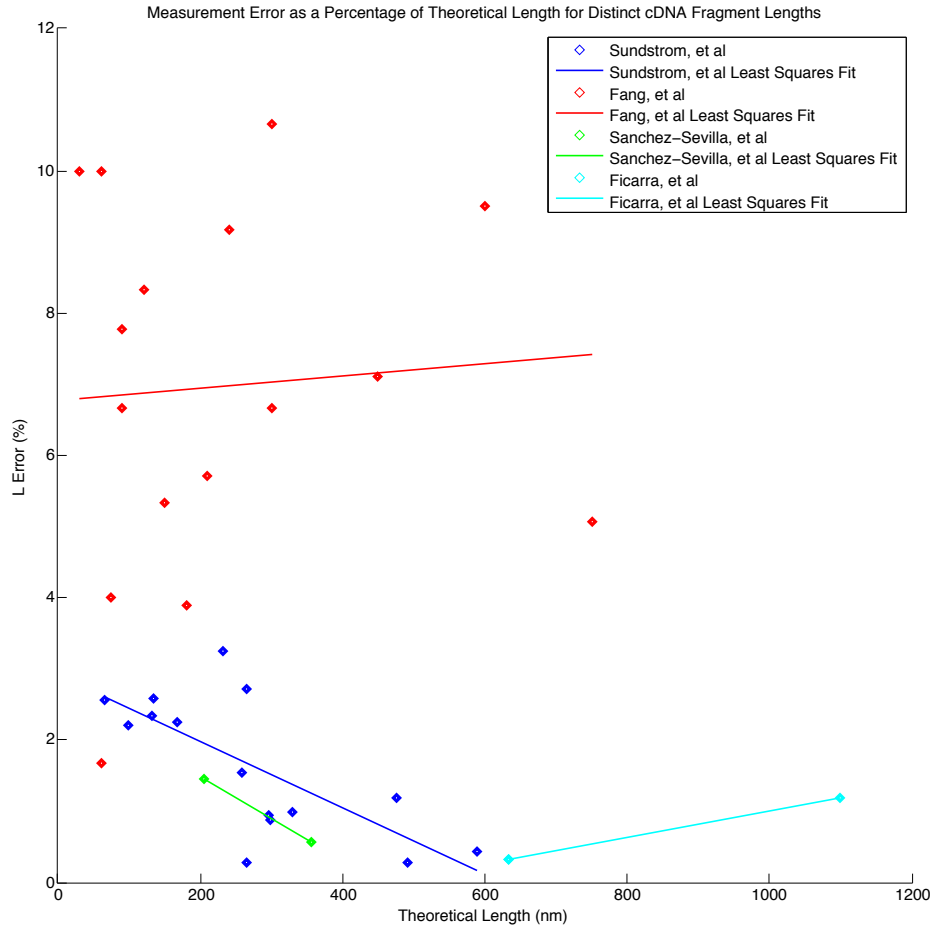


Fig. 9: Errors of measured lengths for distinct cDNA fragment lengths across cited studies, expressed as a percentage of their respective theoretical fragment lengths.

TABLE 2: Automated DNA sizing accuracy claims. Columns give the authors of the studies, the theoretical DNA fragment lengths under investigation (τ) in nanometers, the errors in nanometers of their best estimator ($\mathcal{L}$) obtained in their respective experiments, the theoretical DNA fragment lengths under investigation (τ) in base pairs, and the errors in base pairs of their best estimator ($\mathcal{L}$) obtained in their respective experiments. Errors 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 0.33 nm = 1 bp. Results from the "Error (%)" column are plotted in Supplementary Figure 9.

Author (year)	**Fragment Length (nm)**	**Error (nm)**	**Fragment Length (bp)**	**Error (bp)**	**Error (%)**
Fang, *et al* (1998)	30.00	3.00	90.91	9.09	10.00
	60.00	1.00	181.82	3.03	1.67
	60.00	6.00	181.82	18.18	10.00
	75.00	3.00	227.27	9.09	4.00
	90.00	6.00	272.73	18.18	6.67
	90.00	7.00	272.73	21.21	7.78
	120.00	10.00	363.64	30.30	8.33
	150.00	8.00	454.55	24.24	5.33
	180.00	7.00	545.45	21.21	3.89
	210.00	12.00	636.36	36.36	5.71
	240.00	22.00	727.27	66.67	9.17
	300.00	20.00	909.09	60.61	6.67
	300.00	32.00	909.09	96.97	10.67
	450.00	32.00	1363.64	96.97	7.11
	600.00	57.00	1818.18	172.73	9.50
	750.00	38.00	2272.73	115.15	5.07
Sanchez-Sevilla, *et al* (2002)	206.00	3.00	624.24	9.09	1.46
	355.00	2.00	1075.76	6.06	0.56
Ficarra, *et al* (2005)	633.40	2.10	1919.39	6.36	0.33
	633.40	2.00	1919.39	6.06	0.31
	1098.00	13.00	3327.27	39.39	1.18
Sundstrom, *et al* (2012)	66.00	1.69	200.00	5.12	2.56
	99.00	2.19	300.00	6.64	2.21
	132.00	3.09	400.00	9.36	2.34
	135.30	3.49	410.00	10.58	2.58
	167.80	3.79	508.48	11.48	2.26
	231.00	7.49	700.00	22.70	3.24
	258.70	3.99	783.94	12.09	1.54
	264.00	7.19	800.00	21.79	2.72
	265.00	0.75	803.03	2.27	0.28
	297.00	2.79	900.00	8.45	0.94
	299.00	2.65	906.06	8.03	0.89
	330.00	3.29	1000.00	9.97	1.00
	475.60	5.65	1441.21	17.12	1.19
	492.40	1.39	1492.12	4.21	0.28
	588.10	2.55	1782.12	7.73	0.43