S.1 Computation of coarse-grain approximations to the tangent

The unit tangents $t_i^{[k]}$ (where for simplicity we only take $k$ odd) to the straight lines that are the best least squares linear approximation to a consecutive run of $(k + 1)$ base-pair locations $r_i, \ldots , r_{i+k}$ can be computed for any configuration, $t_i^{[k]}$ as follows. First calculate the (geometrical) centre of mass $c_i^k = (\sum_{j=i}^{i+k} r_j)/(k + 1)$. Then $t_i^{[k]}$ is the unit eigenvector (with positive projection on the chord $r_{i+k} - r_i$) corresponding to the largest eigenvalue of the (local gyration) matrix $\sum_{j=i}^{i+k} (r_j - c_i^k) \otimes (r_j - c_i^k)$. The case $k = 1$ reduces analytically to the unit tangent to the junction chord between two consecutive base pair origins $t_i^{[1]} = (r_{i+1} - r_i)/\|r_{i+1} - r_i\|$, while nonlocal coarse grain choices $k > 1$ must be computed numerically.

S.2 Details regarding the cgDNAmc code

S.2.1 Downloading the software

The C++ code cgDNAmc, along with two libraries it depends upon, algebra3d and cgDNArecon, is freely available with online instructions on how to download, compile, and run it\footnote{\url{http://www.doornik.com/research/ziggurat.pdf}}. The user has to supply any desired problem-specific, post-processing code fragments implementing specialised techniques such as the sliding-window average used in modelling cryo-EM experimental data.

The remainder of this section describes our Monte Carlo implementation in further detail. The simulations described here are not particularly intensive, nevertheless we have taken some efforts to make cgDNAmc code efficient. Benchmark results presented below were obtained on a mid-range laptop computer.

S.2.2 Direct Monte Carlo sampling

As described in the main text, a key step in our direct Monte Carlo sampling is the Cholesky decomposition $K = LL^T$ of the sparse stiffness matrix $K$. For efficient sampling, the key property of the Cholesky factorization is that if $K$ has bandwidth $m$ (meaning that all nonzero entries are within $m$ rows of the diagonal, so for us $m = 17$), then $L$ also has bandwidth $m$\footnote{http://arxiv.org/abs/1404.0390}. After this step, a new energy $E(y) = \frac{1}{2} y^T y$ with $y = L^T (w - \hat{w})$ yields a probability density function on $y$ that is the product of uncoupled univariate normal distributions:

$$p_y(y) = \prod_{i=1}^{12n-6} \left( \frac{\beta}{2\pi} \right)^{\frac{1}{2}} e^{-\frac{\beta y_i^2}{2}},$$

(S.1)

To make a single draw $y$ from this distribution, each component $y_i$ is taken as a random number from the normal distribution with mean 0 and standard deviation $\beta^{-\frac{1}{2}}$. Note that units of the stiffness matrix $K$ in the cgDNA model are such that $\beta = 1$. For the sake of efficiency uniform deviates are generated using the xorshift1024* implementation\footnote{http://arxiv.org/abs/1404.0390} of the xorshift algorithm\footnote{http://www.doornik.com/research/ziggurat.pdf} and are subsequently converted to normal deviates using the ZIGNOR implementation\footnote{http://www.doornik.com/research/ziggurat.pdf} of the Ziggurat algorithm\footnote{http://www.doornik.com/research/ziggurat.pdf}.

The draw of the internal coordinates $w$ corresponding to $y$ is obtained from the equation $y = L^T (w - \hat{w})$ by solving $L^T z = y$ for $z$ (taking advantage of the upper triangular, banded structure of $L$ using an appropriate solver from LAPACK\footnote{http://www.doornik.com/research/ziggurat.pdf}) and then setting $w = z + \hat{w}$.

An alternative approach to obtain direct sampling would involve a spectral decomposition of $K$ in place of the Cholesky factorisation, i.e.

$$K = PD\hat{P}^T,$$

(S.2)

with $P$ orthogonal and $D$ diagonal. Here a similar change of variable $y = D^{\frac{1}{2}} P^T w$ can be used so that $w = PD^{-\frac{1}{2}} y$.

This has been successfully exploited by Czapla et al.\footnote{http://arxiv.org/abs/1404.0390} for the case where $K$ is block diagonal. However in our
setting with a (potentially large) banded $K$ that approach is significantly less efficient, since the matrix $PD^2$ would not be sparse, and a dense matrix-vector multiply must be carried out in the construction of each draw. To give an example a simulation calculating $(t_0[0] \cdot t_0[0])$ for 1 million configurations of the $\lambda_3$ sequence of length 300 bp using Cholesky decomposition takes just above 3 minutes, while using spectral decomposition the running time is around 2 hours.

S.2.3 Metropolis Monte Carlo sampling

As described in the main text, to sample the non-Gaussian distribution (11) we use the Metropolis algorithm (see [6] for a treatment similar to that we use here). Given a prior configuration with internal-variable vector $w$, we follow the direct Monte Carlo procedure from the previous section to generate a new draw $w^*$ for a treatment similar to that we use here. Given a prior configuration with internal-variable vector $w$, we accept or reject it as follows: if $J(w^*) \geq J(w)$, we accept $w^*$; whereas if $J(w^*) < J(w)$ we accept $w^*$ with probability $J(w^*)/J(w)$ and otherwise reject it (in which case we append a new copy of $w$ to our ensemble). This acceptance criterion is one way of ensuring the crucial property of detailed balance, which requires that

$$
\alpha(w \rightarrow w^*) P(w \rightarrow w^*) \tilde{p}_w(w) = \alpha(w^* \rightarrow w) P(w^* \rightarrow w) \tilde{p}_w(w^*),
$$

where $\tilde{p}_w$ is the probability density function (11), $\alpha(y \rightarrow z)$ is the conditional probability density in our Metropolis algorithm for choosing state $z$ given prior state $y$ (which in our scheme is independent of $y$ and equals $p_w(z)$ from (11)), and $P(y \rightarrow z)$ is the probability in our Metropolis algorithm of accepting the new state $z$ given a prior state $y$ (which in our scheme is 1 if $J(z) \geq J(y)$ and $J(z)/J(y)$ otherwise).

The efficiency of any Metropolis method depends strongly on the acceptance rate for the given move set, which can be puntingly small. In the particular case of the pdf (11) with the explicit choice (12) for $J$, and the cgDNA energy (10), the observed acceptance rates depend on the length of the simulated oligomers. For oligomers of 300 bp the acceptance rate is approximately 37%, which is perfectly acceptable. For oligomers 5 times as long (1500 bp – as used for computing the Flory persistence vectors) the acceptance rate drops to just under 5%, with a corresponding increase in the number of draws required to obtain convergence.

S.2.4 Rigid base pair marginals

We remark that many expectations of interest involve only the inter part of the configuration variable $w$ so that the number of degrees of freedom can be reduced by one half by computing the marginal distribution for the inter variables. As the original distribution is Gaussian its marginals are also Gaussian, but the resulting marginal stiffness matrix is now dense, so that sparse computations can no longer be used. As a consequence a calculation of $\langle 0 \rangle$ for 1 million configurations of the $\lambda_3$ fragment using the marginal distribution takes around 23 minutes, nearly 7 times slower than generating ensembles in the full $w$ space and discarding all the intr variables.

S.2.5 Reconstruction of 3D shapes

The first step in calculating our observables is reconstructing a 3D shape of a molecule from a given internal coordinate vector $w$ as detailed in [7]. As mentioned in the previous section, the calculation of tangent-tangent correlations, arclengths and Flory vectors require only the inter part of $w$. As a result we only reconstruct base pair positions $r_i$ and orientations $R_i$, which takes only half the time of reconstructing a full 3D configuration of rigid bases. The reconstruction procedure, implemented by the cgDNArecon library, involves evaluating half rotations, composing rotations, applying rotations to vectors and adding vectors. A careful numerical study of efficiency of different parametrisations of rotations (namely Cayley vectors, unit quaternions and rotation matrices) using the algebra3d library has been performed. An explicit half-rotation formula for unit quaternions proved to be 60% faster than a similar formula for Cayley vectors. (For rotation matrices, the analogous calculation would require, e.g., an iterative algorithm of computing the principal square root and so was not considered). As expected, for composition of rotations, quaternion multiplication was faster than matrix multiplication, with our observed difference being 20%. On the other hand, in the case of applying a rotation to a vector, the matrix-vector product was 5 times faster than a specialised quaternion multiplication. In fact the fastest way to apply a rotation given as unit quaternion to a vector was to convert the quaternion to a rotation matrix first (this takes only twice the time of the matrix-vector product). Efficiency of converting between all three parametrisations was also analysed. This suggested, for example, that the formula for computing a rotation matrix
for a given Cayley vector of \([7]\) is two times slower than conversion of a Cayley vector to quaternion and subsequent conversion of the quaternion to a rotation matrix.

Considerations similar to the above suggested two approaches to the reconstruction procedure. The first one uses directly the Cayley vectors of the configuration variable \(w\) to calculate half rotations and converts to rotation matrices for all subsequent calculations. The other one, that finally proved to be 30% faster, begins with converting the Cayley vectors to quaternions, then computes half rotations using quaternions, and finally converts quaternions to matrices when rotations need to be applied to vectors.

### S.2.6 Remarks on parallelisation

We first note that in \(cgDNAmc\) pseudo-random numbers are generated sequentially to ensure reproducibility of results. Also the reconstruction procedure is inherently sequential. The conversion of the decoupled normal deviates \(y\) to an internal coordinate vector \(w\) depends on the underlying LAPACK routine, that might already be optimized to use available multiple cores, but the \(cgDNAmc\) code has no other explicit parallelisation. In part this is because each configuration can be generated and analysed independently of all others, so that the suggested solution for generating large ensembles is to run multiple independent simulations at the same time, with a different seed for the pseudo-random number generator in each instance. By linearity, expectations from multiple runs can be aggregated as a weighted average with weights proportional to the number of configurations generated in each independent run. As an example we achieved a 2.4 speed up in this way by running four independent threads on a single laptop.

### S.2.7 Run-times of key steps of algorithm

A simple profile of run times for the key steps of a simulation that calculates five expectations using 1 million configurations of the 300 bp \(\lambda_3\) oligomer is:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Run time [s]</th>
<th>% of simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation of (y)</td>
<td>59.08</td>
<td>12.66%</td>
</tr>
<tr>
<td>Transformation to (w)</td>
<td>74.75</td>
<td>16.02%</td>
</tr>
<tr>
<td>Shape reconstruction</td>
<td>41.41</td>
<td>8.87%</td>
</tr>
<tr>
<td>Calculating (\langle t_0^0 \cdot t_1^0 \rangle)</td>
<td>6.63</td>
<td>1.42%</td>
</tr>
<tr>
<td>Calculating (\langle t_0^{[1]} \cdot t_1^{[1]} \rangle)</td>
<td>273.28</td>
<td>58.55%</td>
</tr>
<tr>
<td>Calculating (\langle s_i^{[1]} \rangle)</td>
<td>3.72</td>
<td>0.80%</td>
</tr>
<tr>
<td>Calculating (\langle s_i^{[11]} \rangle)</td>
<td>0.61</td>
<td>0.13%</td>
</tr>
<tr>
<td>Calculating Flory vecs</td>
<td>6.12</td>
<td>1.30%</td>
</tr>
<tr>
<td>Other</td>
<td>1.14</td>
<td>0.24%</td>
</tr>
</tbody>
</table>

The entire simulation run time is 466.74 s or 100.00% of the total.

The time necessary to evaluate most of the expectations is a negligible fraction of the total, except for the generalized-chord expectation \(\langle t_0^{[11]} \cdot t_1^{[11]} \rangle\), where the computation of the principal eigenvector of the local gyration matrix is quite costly.

### S.3 DNA sequences

#### S.3.1 \(\lambda\) phage genome

Five fragments of length 300bp drawn from the \(\lambda\) phage genome of Sanger et al. [8]. The full sequence is available [online]\(^4\). A single repeat was used for \(\ell_p\) computations, 5 repeats for \(\ell_F\).

\(\lambda_1\) (base pairs 25201 – 25500)

\[
\begin{align*}
\text{TTGTAGGCTC} & \quad \text{AAGAGGGTGT} \\
\text{GTCCGTGCT} & \quad \text{AGGTAATAA} \quad \text{CTGACCTGTC} \\
\text{GACCTAATA} & \quad \text{TTCTATATTG} \quad \text{TGTTCTTTTC} \\
\text{TGCAAAAAAG} & \quad \text{TGGGGAAGTG} \\
\text{AGTAATGAAA} & \quad \text{TTATTTCTAA} \quad \text{CATTTATCTG} \\
\text{CATCATACCT} & \quad \text{TCCGAGCATT} \\
\text{TATTAAGCAT} & \quad \text{TTCGCTATAA} \quad \text{GTTCTGCTC} \\
\text{TTGTTCTTTC} & \quad \text{GAAGAGTGTG} \quad \text{TTTTTCATT}
\end{align*}
\]

GTACTTTACC TTCATCTCTG TTCATTATCA TCGCTTTTAA AACGGTTCGA
CCTTCTAATC CTATCTGACC ATTATAATTT TTTAGAATGG TTTCATAAGA

λ3 (base pairs 21901 – 22200)
CGTTAACGCT GCGGGTAACG CGGAAAACAC CGTCAAAAAC ATTTGCAATTA
ACTATATTGT GAGCTCTGCA TAATGGCATT CAGAATGAGT GAAACACACG
GGACCATAAA AATTAAAAAT CTGCTGCGCC GAACCTATGA ATTTATGCTG
GAAAGTGACG CATATTACCC GCCCTACACC GCTCTGCGTG CAATACAGTC
CTGATGAGCC ATCTGAGCAT CTCGTTGAGA ACCATCGGGG TAAACCGGT

λ3 (base pairs 36901 – 37200)
TAGAGCGATT TACATTCTGA ACCAGACTCT TGTCATTAGT TTTGGTAAGG
AGAAAATTT TTTCCATCGAT TTTATGAATA TTTAAATAAT TGGGAGCAAC
CTGCGGTTGA CAGTTATTAC GCAGCGAGAA ATTAAGGAAA ACAGACAGGT
TTATCTGAGG CTTAATTTCT CGTTATTTTT TGTCTGCGCTA AGTCGCAATA
AAACCTAATCT CATAATGATG CTTGAATTTA GCTAAGCGGA GTGAAAATTC
CTCTAAATGCG AACTGAAGCTTAC TTGGCAGCTGATTCTAAAA TGTATTCAGT

λ4 (base pairs 24301 – 24600)
CTATGACTCTG ACCGCACTCTG CTTCACTTGTC GGCTAGTTGC GGAGCGGACA
TTACAAACCTG GGCTTGCCTGG GCAGCCACCT GTTGCCAAATG ACCTGCGCTG
GAATTGGCGGA CAGGTGCTTTT GAATGGAAGAC CCGCCATATG CCGGGTTATG
ATAAAAAGTAT CTGGTTCGCT TGCGATAAAC GCTGTAATTT TCTATTTTCT
ATCATTATCT AGATCCCTCTG GCAGAAATCT GCAGATGCTTG ATGACGACTG
ATACATATAT CTTTATCAAT AATTTGGGGA ATGGGACAGTAA TGTATACCAG

λ5 (base pairs 37801 – 38100)
CCTGACTGCC CCATCCCCAT CTTGTCTGCG ACAGATTCCT GGGATAAGCC
AAGTTTCATTT TTCTTTTTTT CTAAATTTGC TTTBGGGAAG ATGGGCTCTG
CAAGCTGCTC TTGTGTTAAT GGTGTTTCTT TGTGCTCAT ACGTTGTCAG
TGACATGCTT GATAATAGT TGGTCTGATG AGGAGGTTTG ATGGAACACG
GCATACCTAT GAAAGATTTA GCAGGTTTCTT TTGGCCAAAC CAAGCAGGT

S.3.2 Virstedt et al. sequences
Sequences used in the experimental study by Virstedt et al. [9]. A single repeat was used for \(\ell_p\) computations, 8 for \(\ell_F\).

\(\gamma_1\) (CA) – 170 base pairs
GAGGATTTCTT GGGAAAACCC TGGTGAGAACAC ACACACATCT ATGAGTACAC
ACACATCTTG CCATCTGACGC ACACACCATC ACACTAGTGAG AAGGCTACAC
ACACAACTGC CCAACATCTG CACACACATC CACACACACAC CACACCACTG
CAGGAAACAG CTTCGACTTC

\(\gamma_2\) (CAG) – 181 base pairs
GAGGATTTCTT GGGAAAACCC TGGGAGGACGC CAGCAGCAAC AGTAGTACAA
GCGAGAGCAC TAAACGACGC AGCAGAAATG GCAATAGAA GAGCAGAGAA
AGCAGAATG CAGTACGAG CAGCAGAAG TACACAAACACA ACAACACGAG
CAGCAGATCC ACAGGAAAAA GCTTCGACTTC
γ3 (NoSeq) – 195 base pairs
GAGGATTCCT GGGAAAACCC TGGCGCAAGA CCGAGTTACT AAACAGGACT
ATTACTGCGA CGCGAAATTGT AGCGCGCAGC CAGTCTCTG CTCACCACCTA
TGCTCTTGTT GCAGCTATTG TACATTAGCT ATCCCTGTTA GCTATAACCTA
CTGATGCTCA ATTACCCGCC TCACACAGGA AACAGCTCGG TCCTC

γ4 (TATA) – 176 base pairs
GAGGATTCCT GGGAAAACCC TGGCGAGGTC TATAAGCGTC TATAAGCGTC
TATGAACGTC TATAAACGTC TATAAACGCC TATAAACGCC TATAAACGCC
TATACCAAGCC TATAAACGCC TATACACGTC TATAGCAGAAC TATACACGTC
TTCACACAGG AAACAGCTCG GTCCTC

S.3.3 Bednar et al. [10] γ5
A sequence designed to be intrinsically straight. 9-15 repeats were used for \( \ell_p \) and \( \ell_w \) computations, 75 for \( \ell_F \).
ATCTAATCTA ACACAACACA

S.3.4 Kahn and Crothers [11] c11t15/γ6
An intrinsically bent sequence with phased A-tracts originally used for minicircle experiments. 2 repeats were used for \( \ell_p \) computations, 10 for \( \ell_F \).
GATGAATTCA CGGATCCGGT TTTTTGCCCG TTTTTTGCCG TTTTTTGCCC
GTTTTTGGCC GTTTTTGGCC GTTTTTCCC GATCGTGAC AGGAATCTCA
GACCTAGGTC GCCTATAATG TGGACCTAAT CATTAACATCAT GCCTGTCGCC
ATGGAATC

S.3.5 Geggier and Vologodskii [12] sequences
Sequences used in the experimental study by Geggier and Vologodskii as provided in their Supplementary Information. A single copy was used for \( \ell_p \) computations.

ACAT – 201 base pairs
AGCTTACACA TATATAACACA TACATATAACACATA CACATATACTGGAGACAT
ACACATATAC AGACATACAT ACAGCATATAC ATATATACAGATACACATAT
ACATACATATA CACATACATAC ATATATACAGCATACACAT ACATACATAC
ACACATACAC ATACATACAC ATACATACAC ATACATACAC ATACATACAC
A

ACCAGG – 201 base pairs
AGCTTACAG GAGGACCCACC AGGACCACCA CCAGGAGGAG CTGCGAGACCA
GGACACCCAG GAGGACCCAG AGGAGGACC CCACACCGAC CACGGACGAC
AGGGAGGAGG CCACAGAGGAC CACAGGAGG AGGACCCACCA GGAGGAGG
GACCCACCC AGGAGGAGG AGCACCAGA CCAGGAGGAC CAGGAGGACC AGGACCACCA
A

ACGAGC – 199 base pairs
AGCTTACGAC GAGGACCGAC AGAGGACGCA GCAGCGCGCT GCAGGACGAC
AGGAGGAGG GCAGGCAGGC ACAGGCAGGC AGGAGGAGG ACAGGAGGAC
AGGAGGAGG GACGAGACCA GCAGGACGAC CAGGAGGACG AGGAGGAGG
CGAGCAGCAG CAGGACGACG AGGACACCGA GCAGCAGGAC CAGGAGG
A
AGAT – 200 base pairs
AGCTTAGAGA TATATAGAGA TAGATATAGA GAGAGATATC TGCAGAGATA
GAGATATATA GAGATAGATA GAGAGATATC TGCAGAGATA
GATAGATATA GATAGATAGA TAGATAGATA GATATATATA
AGATAGAGA TAGATAGAGA GATATAGATA TAGATAGATA TATAGATATA

AGC – 199 base pairs
AGCTTAGCAG CAGCAGCAGC AGGCTGAGGA GCAGCAGCAG
CAGCTAGCAG CAGCAGCAGC AGGAGCAGGA GCAGCAGCAG
GCTAGCAGCA GCAGCAGCAA GCAGCAGCAA GCAGCAGCAA
TAGCAGCAGC AGCAGCAGCAA GCAGCAGCAA GCAGCAGCAA

CAAA – 200 base pairs
AGCTTACAAAC AACAACACAC CAACACACAA CAAACAACCAC
CAACACACAA CAACACACAA CAACACACAA CAACACACAA
CAACACACAA CAACACACAA CAACACACAA CAACACACAA
CAACACACAA CAACACACAA CAACACACAA CAACACACAA

CAA – 198 base pairs
AGCTTCAGTG AAGCTAGTG TGGAGCTGG TAGCTGGCAG
AGGTATAGTG AGGTATAGTG TAGCTGGCAG TGGAGCTGG

CAGT – 200 base pairs
AGGTTCAGTC ACTACATCAT CATCTACTAC ATCTACATCA
CTACATCAT ACTACATCAT CATCTACTAC ATCTACATCA
CATCTACTAC ACTACATCAT CATCTACTAC ATCTACATCA
CATCTACTAC ACTACATCAT CATCTACTAC ATCTACATCA

CATCTA – 200 base pairs
AGCTTCATCT ACTACATCAT CATCTACTAC ATCTACATCA
CATCTACTAC ACTACATCAT CATCTACTAC ATCTACATCA
CATCTACTAC ACTACATCAT CATCTACTAC ATCTACATCA

HPL1 – 198 base pairs
AGCTTTAGTA GCCTAGTGAC CTAGAGCTTC GCCTAGCTTC
CCAGATTCGC ATCTCGTAAG AAGGCTGAGGA TCGGAGAC
GATGTCAGTG TAGCTGAGGA TAGCTGAGGA GCAGGAGGA
TAGCTGAGGA TAGCTGAGGA GCAGGAGGA GCAGGAGGA

LPL1 – 200 base pairs
AGCTTTAGTA GCCTAGTGAC CTAGAGCTTC GCCTAGCTTC
CCAGATTCGC ATCTCGTAAG AAGGCTGAGGA TCGGAGAC
GATGTCAGTG TAGCTGAGGA TAGCTGAGGA GCAGGAGGA
TAGCTGAGGA TAGCTGAGGA GCAGGAGGA GCAGGAGGA
HPL2 – 198 base pairs
AGCTTACGAC GAACGACGAC GAACGACGAA CGAACGACGA ACGCTGCAGA CGACGAACGA CGAACGACGA CGAACGACGA ACGACGAACG ACGACGAACG ACGACGAACG AGACGAACGA ACGACGAACG AACGACGACG AACGAACGAC GAACGACGA

LPL2 – 201 base pairs
AGCTTGCATA GCCATTAGCC ATGCATAGGC ATATGGCATT AGGCACTGCA GGCCCATAGG CATGCATAGG CATAGGCCAT GGCATAGGCA TTAGGCATGC GATAGGCATAG GCATGGCATA GGCATTAGGC ATGCATAGGC ATAGGCATGG CATAGGCATT

SG1 – 199 base pairs
AGCTTAGGAC TACGAACGCT AGCTTAGCTA CCAGCGAGTA CACTGCAGCA GCAGCTAGCT AGCGCGATGC CCAGCTGAGA TCGACGATCG ATGGCGATTA TCAGCTAGCA GCTAGCGATC GACGCGCGAT GCGCAGCTGA GCTAGCTGAT

λ6 – 205 base pairs
AGCTTCTCCT TTGATGCGAA TGCCAGCGTC AGACATCATA TGCAATGACT CACCTGCATC CTGAACCCAT TGACCTCCA CCCCCTATAA GGCGATCGTA ATGATGTCGA TAGCTATTAA GCGGCTCTGT TGAGATATCT GCAGCAAGAAA CTTCCAGGG TCCAGGACTG AAGACGAGTC TTCCAGGAT GGCGA

γ7 (IS) – 211 base pairs
CTAGAGTCTC AGGCTAGCGT CGACCATAGC ACTGACTCAGT AGACTGACGTG ACTGACTCAGT ACTGACTCAGT ACTGACTCAGT ACTGACTCAGT ACTGACTCAGT
S.4 Supplementary Figures

S.4.1 Sequence is significant—some specific cases

Figure S1: Ground state configurations and Flory persistence vectors for various DNA sequences (an interactive version of Figure 4 of the main text). The columns show: (left) the six distinct poly-dinucleotide sequences, (middle) the six selected λ-phage fragments λj, and (right) the seven sequences γj. The first row of panels shows visualizations of the shapes of cgDNA ground state configurations, while the second row shows plots of Flory persistence vectors $\overline{\Pi}_2$ for the Gaussian $\overline{\Pi}_1$ (solid) and perturbed $\overline{\Pi}_2$ (dashed) ensembles. (All six panels are U3D, so that with the appropriate viewer, e.g. Acrobat Reader V7 or higher, they can be interactively rotated and magnified.)
S.4.2 Sensitivity to Jacobian perturbation

We used the two sequences poly(G) and $\lambda_3$ to explore the sensitivity of the persistence length values $\ell_p(S)$ on the inclusion of the Jacobian factor from Eq. (12) in the probability density function, as seen in Eq. (11). Results are presented in Fig. S2. Some difference is perceptible, including a difference in the period of the small oscillations in the case of poly(G), but the magnitude of these effects is rather small at these length scales.

Figure S2: Sensitivity of tangent-tangent correlation data to inclusion of Jacobian factor. Direct Monte Carlo simulation (which does not use the Jacobian) in black, Metropolis Monte Carlo (which incorporates Jacobian) in red; left panel is for 300 bp poly(G) fragment (173 bp – direct, 175 bp – Metropolis) and right panel is for $\lambda_3$ (162 bp – direct, 167 bp – Metropolis). In each case 10 bp were excluded at each end.
S.4.3 Monte Carlo convergence

In the main document, we report two types of convergence results for the Flory persistence vector, first that $10^5$ MC draws is a sufficiently large number of samples, and second that 1.5 Kbp is a sufficiently long sample to yield an overall standard error of less than 0.5 nm. For the apparent persistence length $\ell_p(S)$, we similarly report that $10^5$ MC samples are sufficient for an accuracy of 0.5 nm (using $j = k = 11$). These convergence conclusions are illustrated in Fig. S3.

Figure S3: Examples of convergence of direct MC sampling (left column) and Metropolis MC (right column) for $\lambda_3$. In the top two panels the curves show the norm of the Flory vector (averaged over MC samples of different sizes) plotted against base-pair number; in all cases, the asymptotic values appears to have been reached by 1.5 Kbp (five repeats of $\lambda_3$). The error bars give the standard error obtained for ten independent MC runs plotted every 100 bp; in the case of direct MC with $10^5$ MC samples, the error bars shrink below $\pm 0.5$ nm, however for Metropolis MC as many as $3 \times 10^6$ accepted configurations (with acceptance rate of 4%) are required for the same accuracy. In the bottom two panels, we show the last 50 bp of the tangent-tangent correlation plot relevant for computing $\ell_p$ of a single repeat of $\lambda_3$ with coarse grain parameters $[j,k] = [11,11]$ (averaged over MC samples of different sizes). The error bars, (plotted every 5 bp) give the standard error obtained for ten independent MC runs. The values of $\ell_p$ extracted from these tangent-tangent plots are 56.5 nm (for direct MC) and 57.5 nm (for Metropolis MC with 37% acceptance rate), both with at least 0.5 nm accuracy for $10^5$ or more accepted MC samples.

S.4.4 Coarse Graining sensitivity

This section provides data to justify assertions made in the main text concerning the dependence of $\ell_p^{[j,k]}$ on the choice of coarse graining parameters $[j,k]$. Figure S4 illustrates some of the tangent-tangent correlation data which is fit to extract $\ell_p^{[j,k]}$. Numerical values of persistence lengths for further cases are presented in Table S2.
Figure S4: Left column. The data $\ln \langle t_i \cdot t_0 \rangle$ for fitting $\ell_p^{[j]}(S)$ with different coarse-graining approximations of tangents. Each panel shows the cases $j = 0, 1, 11, 21$ for each of the three sequences poly(G) (top), poly(A) (middle), $\lambda_3$ (bottom). The cases $j = 0, 11, 21$ yield very similar plots, whereas $j = 1$ shows some significant deviation with much larger oscillation and an overall displacement downward, meaning that in the estimation of persistence length the gradient of the linear best fit is quite sensitive to whether or not the line is assumed to pass through the origin. The plots for $j = 0$ and $j = 11$ exhibit oscillations of roughly the same magnitude, presumably reflecting an alignment of the base pairs with the local axis of the local helical structure. In addition, the initial behaviour for the first few base pairs is notably different for the choices $k = 0, 1, 11, 21$ (which is entirely reasonable given the block structure of the stiffness matrix in the cgDNA model) which leads to the consistent (approximate) ordering $1 < 0 < 11 < 21$.

Right column. The data for fitting the dimensional persistence lengths $\ell_p^{[0,k]}(S)$ for different coarse-graining choices of arc length. Each panel shows the cases $k = 1, 11, 21$ for each of the three sequences poly G (top), poly A (middle) and $\lambda_3$ (bottom). For poly(G) there is a difference of 6.4 nm between $k = 1$ and $k = 11$ while for poly(A) the same difference is only 3.0 nm.
S.4.5 Sequence-dependence of persistence lengths

Figure S5 below is the analogue of Fig. 3, which shows histograms of $\ell_F$ (in nm) and $\ell_p^{[0,0]}$, and $\ell_d^{[0,0]}$ (in bp units), but now for $\ell_p^{[11,11]}$ and $\ell_d^{[11,11]}$ in nm.

Figure S5: Normalised histograms of persistence lengths $\ell_F$ in green, $\ell_p^{[11,11]}$ in blue, $\ell_d^{[11,11]}$ in red (all in nm units) for 220 bp fragments from $\lambda$-phage (left) and with random sequence (right). In addition, in each panel, the associated persistence lengths for the six distinct poly(dinucleotide) sequences are marked with coloured circles, with two circles per sequence as $\ell_p$ and $\ell_d$ almost coincide for these almost straight sequences. The harmonic means of $\ell_F(S_j)$ for the $\lambda$ and random ensembles are respectively 55.7 nm and 55.6 nm, of $\ell_d(S_j)$ 59.5 nm and 58.8 nm, and of $\ell_p(S_j)$ 53.2 nm and 53.5 nm.
S.4.6  Sequence-averaged persistence lengths

Figure S6 shows the tangent-tangent correlation plots that were used to compute $\bar{\ell}_d$ and $\bar{\ell}_p$.

Figure S6: Tangent-tangent correlation plots used for extracting $\bar{\ell}_d^{[0,0]}$ and $\bar{\ell}_p^{[0,0]}$ in bp units (left panel) and $\bar{\ell}_d^{[11,11]}$ and $\bar{\ell}_p^{[11,11]}$ in nm units (right panel). As per the definitions of these quantities, each blue curve is the log of the average over sequence and over MC samples of $t_{ij} \cdot t_{0j}$, while each red curve is the log of the average over sequence and MC samples of the ratio $(t_{ij} \cdot t_{0j})/(\hat{t}_{ij} \cdot \hat{t}_{0j})$ ($j = 0$ in left panel, $j = 11$ in right panel). In the left panel, the factorization that involves dividing by the intrinsic shape term greatly reduces the oscillations apparent in the blue curve. In the right panel, the use of a coarse grain arclength also reduces the oscillations in the blue curve relative to the blue curve in the left panel.
S.4.7 Some tangent-tangent simulated correlation plots for sequences with experimental data

Figure S7: Tangent-tangent correlation data used for extracting $\ell_p^{[11,11]}$ and $\ell_w^{[11,11]}$. Left panel: plots for a sequence with relatively high $\ell_p$ (CAACTT from Geggier and Vologodskii, red) and one with relatively low $\ell_p$ (CAG from Virstedt et al., blue). Middle and right panels: Five plots each for extracting $\ell_w$ for Bednar et al.’s straight molecule (middle) and λ-phage measurement (right). Each of the five plots in each panel represents an average over 25 MC samples and over all possible 1-nm-shifted windows of each given width $\Delta$ up to 40 nm (avoiding the first or last 15 bp). Note the wide variation in curve shape and best-fit slope $-1/\ell_w$ among the five curves in the middle panel, corresponding to the relatively large uncertainty (±15 nm) in Bednar et al.’s reported value of $\ell_w$. The variation is less prominent in the right panel since there are more fragments (37 as compared to 25) and more windows per fragment (as the λ fragments are 300 bp whereas the straight molecule is 180 bp). For our reported values of $\ell_w$ (58 nm and 52 nm respectively), we computed 1,000 such curves and averaged the resulting values of $\ell_w$; Bednar et al.’s reported values would seem to be the result of analysing a single such curve, and for a single window size of 40 nm.
### Tables of Numerical Data

Table S1 provides the numerical values of the persistence lengths extracted from the plots of simulated expectations presented in Figures 4 and 5 in the main text, along with the numerical values of the points in the scatter plot of computed versus experimentally observed persistence lengths shown in Figure 7. Table S2 quantifies the sensitivity of persistence lengths to coarse-grain choices \([j, k]\) in arc-length and tangent fit in the cases of four sequences. Tables S3 and S4 provide the MD and MC numerical simulation data used to compare fits of persistence lengths for short fragments of the six distinct poly-dimer sequences, cf. Figure 6 and Table 4 in the main text.

#### Table S1: Numerical values of persistence lengths

<table>
<thead>
<tr>
<th>Group</th>
<th>Molecule</th>
<th>(\ell_p^{[0,0]}) (bp)</th>
<th>(\ell_d^{[0,0]}) (bp)</th>
<th>(\ell_F) (nm)</th>
<th>(\ell_{F,J}) (nm)</th>
<th>(\ell_p^{[11,11]}) (nm)</th>
<th>(\ell_p^{exp}) (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poly-dinucleotides</td>
<td>poly(AA)</td>
<td>219</td>
<td>221</td>
<td>72.7</td>
<td>70.2</td>
<td>73.5</td>
<td>50.4</td>
</tr>
<tr>
<td></td>
<td>poly(GG)</td>
<td>173</td>
<td>178</td>
<td>56.2</td>
<td>55.7</td>
<td>56.0</td>
<td>41.7</td>
</tr>
<tr>
<td></td>
<td>poly(TA)</td>
<td>146</td>
<td>148</td>
<td>47.2</td>
<td>46.9</td>
<td>47.0</td>
<td>42.7</td>
</tr>
<tr>
<td></td>
<td>poly(AC)</td>
<td>169</td>
<td>173</td>
<td>55.5</td>
<td>56.6</td>
<td>55.4</td>
<td>50.7</td>
</tr>
<tr>
<td></td>
<td>poly(AG)</td>
<td>192</td>
<td>194</td>
<td>64.0</td>
<td>64.5</td>
<td>64.5</td>
<td>52.6</td>
</tr>
<tr>
<td></td>
<td>poly(CG)</td>
<td>166</td>
<td>168</td>
<td>54.9</td>
<td>54.0</td>
<td>55.3</td>
<td>49.7</td>
</tr>
<tr>
<td>Fragments of (\lambda)</td>
<td>(\lambda_1)</td>
<td>99</td>
<td>231</td>
<td>48.4</td>
<td>52.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(\lambda_2)</td>
<td>152</td>
<td>184</td>
<td>55.3</td>
<td>55.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(\lambda_3)</td>
<td>162</td>
<td>186</td>
<td>58.3</td>
<td>59.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(\lambda_4)</td>
<td>171</td>
<td>181</td>
<td>56.2</td>
<td>57.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(\lambda_5)</td>
<td>178</td>
<td>181</td>
<td>59.1</td>
<td>58.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(\lambda_6)</td>
<td>168</td>
<td>178</td>
<td>57.4</td>
<td>57.6</td>
<td>56.9</td>
<td>48.0</td>
</tr>
<tr>
<td>Virstedt et al [9]</td>
<td>(\gamma_1)</td>
<td>166</td>
<td>172</td>
<td>54.9</td>
<td>57.0</td>
<td>54.8</td>
<td>45.5</td>
</tr>
<tr>
<td></td>
<td>(\gamma_2)</td>
<td>155</td>
<td>176</td>
<td>56.0</td>
<td>56.1</td>
<td>51.5</td>
<td>41.7</td>
</tr>
<tr>
<td></td>
<td>(\gamma_3)</td>
<td>169</td>
<td>176</td>
<td>56.1</td>
<td>56.4</td>
<td>56.3</td>
<td>50.5</td>
</tr>
<tr>
<td></td>
<td>(\gamma_4)</td>
<td>153</td>
<td>172</td>
<td>48.3</td>
<td>42.9</td>
<td>51.5</td>
<td>45.5</td>
</tr>
<tr>
<td>Bednar et al [10]</td>
<td>(\gamma_5)</td>
<td>176</td>
<td>179</td>
<td>58.2</td>
<td>58.7</td>
<td>58.2</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>Avg of (\lambda)</td>
<td>134</td>
<td>187</td>
<td>51.5</td>
<td>52.5</td>
<td>52.0</td>
<td>45.0</td>
</tr>
<tr>
<td>Kahn &amp; Crothers [11]</td>
<td>(\gamma_6)</td>
<td>172</td>
<td>175</td>
<td>57.2</td>
<td>57.5</td>
<td>57.4</td>
<td>49.5</td>
</tr>
<tr>
<td></td>
<td>ACAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>51.4</td>
<td>46.0</td>
</tr>
<tr>
<td></td>
<td>ACCAGG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>54.2</td>
<td>47.5</td>
</tr>
<tr>
<td></td>
<td>ACGAGC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>57.9</td>
<td>51.0</td>
</tr>
<tr>
<td></td>
<td>AGC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>57.1</td>
<td>47.0</td>
</tr>
<tr>
<td></td>
<td>CAA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>60.0</td>
<td>50.0</td>
</tr>
<tr>
<td></td>
<td>CAACCTT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>62.5</td>
<td>51.0</td>
</tr>
<tr>
<td>Geggier &amp; Vologodskii [12]</td>
<td>AGAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>51.5</td>
<td>47.0</td>
</tr>
<tr>
<td></td>
<td>CAGT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>60.0</td>
<td>51.5</td>
</tr>
<tr>
<td></td>
<td>LPL1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>51.5</td>
<td>49.0</td>
</tr>
<tr>
<td></td>
<td>HPL1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>58.6</td>
<td>48.5</td>
</tr>
<tr>
<td></td>
<td>LPL2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>51.5</td>
<td>45.5</td>
</tr>
<tr>
<td></td>
<td>HPL2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>57.9</td>
<td>54.0</td>
</tr>
<tr>
<td></td>
<td>SG1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>55.5</td>
<td>48.5</td>
</tr>
</tbody>
</table>

**Table S1:** Numerical values of persistence lengths: Columns 3–6 are the persistence lengths derived from the data plotted in Figures 4 and 5 of the main text (where \(\ell_F\) denotes the Flory persistence length computed from the distribution including the Jacobian factor). Columns 7 and 8 are data used in the comparison between simulation and experimental results shown in Figure 7 of the main text (experimental results taken from citations indicated in each section, with experimental results for poly-dinucleotides and \(\lambda_6\) from [12], see main text).
Table S2: The effect of different choices of coarse-graining parameters \([j,k]\) for tangents and arc lengths on the value of \(\ell_p^{[j,k]}\) and \(\ell_d^{[j,k]}\) for four different sequences of length 300 bp: poly(AA), poly(AT), poly(GG) and \(\lambda_3\).

<table>
<thead>
<tr>
<th>(\ell_p^{[j,k]})</th>
<th>(\ell_d^{[j,k]})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s_0)</td>
<td>(s_1)</td>
</tr>
<tr>
<td>(t_0)</td>
<td>(219)</td>
</tr>
<tr>
<td>(t_1)</td>
<td>(205)</td>
</tr>
<tr>
<td>(t_{11})</td>
<td>(224)</td>
</tr>
<tr>
<td>(t_{21})</td>
<td>(228)</td>
</tr>
</tbody>
</table>

Table S3: Expectations \(\ln(t_i \cdot t_0)\) evaluated on MC and MD ensembles for 18bp fragments containing the six distinct dimer steps. \(t_0\) is the basepair normal to the fourth base pair from one end, and the row index \(i = 1, \ldots, 11\) runs until the fourth basepair from the other end. For formatting reasons, actual values are table entries divided by 100.

<table>
<thead>
<tr>
<th>AA</th>
<th>AG</th>
<th>CG</th>
<th>GG</th>
<th>TA</th>
<th>TG</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC</td>
<td>MD</td>
<td>MC</td>
<td>MD</td>
<td>MC</td>
<td>MD</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>-0.75</td>
<td>-0.82</td>
<td>-0.78</td>
<td>-0.98</td>
</tr>
<tr>
<td>-1.51</td>
<td>-1.48</td>
<td>-1.89</td>
<td>-2.30</td>
<td>-2.29</td>
<td>-1.90</td>
</tr>
<tr>
<td>-2.28</td>
<td>-2.07</td>
<td>-2.68</td>
<td>-3.76</td>
<td>-2.56</td>
<td>-1.96</td>
</tr>
<tr>
<td>-2.96</td>
<td>-2.55</td>
<td>-4.07</td>
<td>-5.41</td>
<td>-5.17</td>
<td>-4.15</td>
</tr>
<tr>
<td>-4.08</td>
<td>-3.69</td>
<td>-5.41</td>
<td>-7.11</td>
<td>-7.41</td>
<td>-6.17</td>
</tr>
<tr>
<td>-4.44</td>
<td>-4.49</td>
<td>-5.42</td>
<td>-6.03</td>
<td>-7.44</td>
<td>-6.81</td>
</tr>
<tr>
<td>-4.80</td>
<td>-4.93</td>
<td>-5.37</td>
<td>-5.42</td>
<td>-6.37</td>
<td>-5.36</td>
</tr>
<tr>
<td>-5.33</td>
<td>-5.36</td>
<td>-5.95</td>
<td>-5.65</td>
<td>-6.99</td>
<td>-6.31</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td></td>
<td>AG</td>
<td></td>
<td>CG</td>
</tr>
<tr>
<td>-----</td>
<td>----</td>
<td>-----</td>
<td>----</td>
<td>-----</td>
<td>----</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-0.11</td>
<td>-0.11</td>
<td>-0.06</td>
<td>-0.17</td>
<td>-0.02</td>
<td>-0.19</td>
</tr>
<tr>
<td>-0.38</td>
<td>-0.28</td>
<td>-0.70</td>
<td>-1.06</td>
<td>-0.93</td>
<td>-0.63</td>
</tr>
<tr>
<td>-0.68</td>
<td>-0.40</td>
<td>-0.92</td>
<td>-1.96</td>
<td>-0.70</td>
<td>-0.20</td>
</tr>
<tr>
<td>-0.90</td>
<td>-0.41</td>
<td>-1.85</td>
<td>-3.16</td>
<td>-2.60</td>
<td>-1.90</td>
</tr>
<tr>
<td>-0.96</td>
<td>-0.37</td>
<td>-1.76</td>
<td>-3.68</td>
<td>-2.39</td>
<td>-1.39</td>
</tr>
<tr>
<td>-0.86</td>
<td>-0.27</td>
<td>-1.98</td>
<td>-3.74</td>
<td>-2.95</td>
<td>-2.41</td>
</tr>
<tr>
<td>-0.62</td>
<td>-0.17</td>
<td>-1.52</td>
<td>-3.35</td>
<td>-3.03</td>
<td>-2.40</td>
</tr>
<tr>
<td>-0.34</td>
<td>-0.09</td>
<td>-0.91</td>
<td>-2.21</td>
<td>-1.55</td>
<td>-1.44</td>
</tr>
<tr>
<td>-0.11</td>
<td>-0.08</td>
<td>-0.51</td>
<td>-1.36</td>
<td>-1.82</td>
<td>-2.00</td>
</tr>
<tr>
<td>-0.02</td>
<td>-0.13</td>
<td>-0.03</td>
<td>-0.30</td>
<td>-0.13</td>
<td>-0.21</td>
</tr>
<tr>
<td>-0.11</td>
<td>-0.22</td>
<td>-0.05</td>
<td>-0.04</td>
<td>-0.25</td>
<td>-0.66</td>
</tr>
</tbody>
</table>

Table S4: Data $\ln t_i \cdot t_0$ analogous to Table S3, but now evaluated on MC and MD ground-state shapes. Again actual values are table entries divided by 100.
References


