PAPER

Check for updates

KOBRA: a fluctuating elastic rod model for slender biological macromolecules†

Robert Welch, [ID] [a] Sarah A. Harris, [ID] [a] Oliver G. Harlen [ID] [b] and Daniel J. Read [ID] *[b]

KOBRA (KirchOff Biological Rod Algorithm) is an algorithm and software package designed to perform dynamical simulations of elongated biomolecules such as those containing alpha-helices and coiled-coils. It represents these as coarsely-discretised Kirchoff rods, with linear elements that can stretch, bend and twist independently. These rods can have anisotropic and inhomogeneous parameters and bent or twisted equilibrium structures, allowing for a coarse-grained parameterisation of complex biological structures. Each element is non-inertial and subject to thermal fluctuations. The speed and simplicity of the algorithm allows KOBRA rods to easily access timescales from nanoseconds to seconds. To demonstrate this functionality, a KOBRA rod was parameterised using data from all-atom simulations of the Ndc80 protein complex, and compared against these simulations and negative-stain EM images. The distribution of bend angles and principal components were highly correlated between KOBRA, all-atom molecular dynamics, and experimental data. The properties of a hinge region, thought to be found at an unstructured loop, were studied. A C++ implementation of KOBRA is available under the GNU GPLv3 free software licence.

1 Introduction

Biomolecules often contain long elongated sections such as α-helices, coiled-coils and DNA/RNA double-helices. These motifs behave dynamically in ways that are much more complex than polymer chain models. They can have non-straight equilibrium structures, and anisotropic, inhomogeneous material parameters. Their global dynamics occur on timescales of microseconds to seconds, and they are frequently found in large, interacting, mesoscale systems, millions of atoms in size.

An example of such a biomolecule is the Ndc80C protein complex (Fig. 1), which forms part of the kinetochore,[1] and which anchors the microtubules of the mitotic spindle to the chromatids during cell division.[2] Ndc80C is comprised mostly of an intrinsically-bent coiled-coil around 60 nm in length, with a hinge region at its centre.[3] The extreme flexibility of many such elongated biomolecules means that we do not possess their fully resolved atomistic structures, however, mesoscale structural information is available from lower resolution methods, such as electron microscopy (EM).[1] To build a model which will enable us to understand the full functionality of the kinetochore, we require a simulation algorithm that can capture how the various components interact.
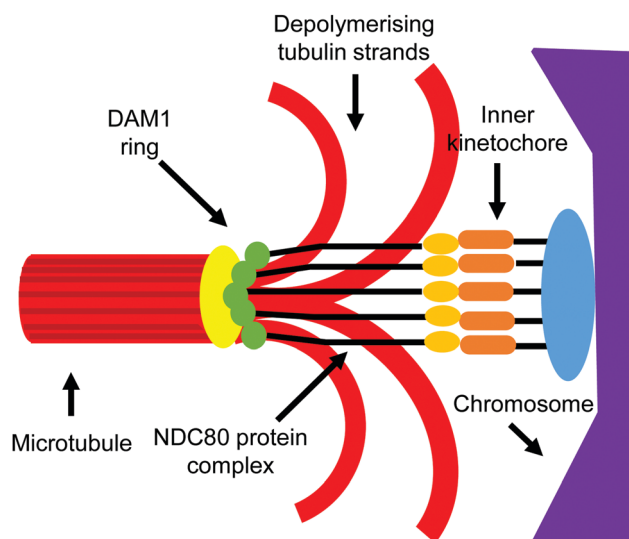
[a] *School of Physics and Astronomy, University of Leeds, Leeds, LS2 9JT, UK*
[b] *School of Mathematics, University of Leeds, Leeds, LS2 9JT, UK.*
  E-mail: D.J.Read@leeds.ac.uk; Tel: +44 (0)113 343 5124
† Electronic supplementary information (ESI) available. See DOI: 10.1039/d0sm00491j

Fig. 1 Schematic diagram of the kinetochore, an example of a mesoscale biological system containing many slender, rod-like structures.[1]

There are few existing algorithms that can help us at this scale. The large size and long timescales of the complete kinetochore system means that an all-atom simulation would be prohibitively expensive computationally. Coarse-grained MD offers a potential solution,[4] however, discrete coarse-grained MD does not scale effectively to systems the size of the kinetochore. As a coarse-grained MD system gets larger, the

Paper

Soft Matter

size of the discontinuities increases, relative to its size, when in reality, the system is approaching a uniform continuum.

The Fluctuating Finite Element Analysis (FFEA) model[5] addresses this problem, by treating large biomolecules as three-dimensional viscoelastic continuum objects. This approximation is valid for biological systems in which large-scale motions are more important than specific chemical interactions. While FFEA is well-suited to globular structures, such as Ndc80C's head and tail, 3D continuum methods are less well-suited to representing long, thin objects, such as Ndc80C's coiled-coil region. Creating trajectories of these objects with correct thermal fluctuations is difficult, as the length scales we need to resolve at are very small compared to the thermal fluctuations.

1.1 1-D rod models for biological objects

In this paper, we present a model specifically designed for dynamical simulations of the full variety of thin biological structures such as α-helices and coiled-coils, using a Kirchoff rod representation[6] and Brownian dynamics.

Elastic rod models are used as coarse-grained models for static modelling of proteins. For example, Wang et al.[7] use a Cosserat model to predict the relationship between strain and chirality, and Neukirch et al.[8] compute coiled-coil pitch and secondary structure. Linka et al.[9] use a Cosserat representation for constitutive models of coiled-coils, but do not perform any explicit dynamics. Here, a Cosserat rod is commonly used to represent the backbone, and attempts are made to try to create a sequences of residues that matches this structure.[10,11]

Elastic rod models are also used to represent DNA. While these models represent DNA as Cosserat curves, they do not perform full dynamical simulations. Manning et al.[12] use Cosserat rods to compute the equilibrium configuration for arbitrary DNA sequences – effectively an energy minimisation. Their more recent work has abandoned rods in favour of using interconnected 2D 'sheets'. Smith et al.[11] use a Cosserat model with anisotropic parameters, but focus almost exclusively on analytical self-contact models for supercoiling and plectoneme formation.

Kirchoff rods[13] are also used for computational models of slender biological objects. Tobias et al.[14] use Kirchoff rods to determine the thermodynamic properties of large ensembles of circular DNA segments. They find exact, analytical equations for the forces and energies in Kirchoff rods, but only for inextensible, uniformly-curved circular rods. Prior et al.[15] develop a Kirchoff rod model for finding the bending and torsional response of helical birods from the bending and torsional responses of the constituent rods.

At larger length scales, Cosserat and Kirchoff elastic rod models are commonly used for macroscopic biological objects. An example is the STRANDS package,[16] which is designed primarily for surgical simulations.

In the next section we set out a general-purpose Kirchoff rod simulation that can incorporate the properties of thin biological objects. Section 3 describes a protocol to set the physical parameters of these rods to match the dynamics of atomistic simulations. Finally, we validate these simulations against both

idealised rod models and experimental and atomistic simulations for the dynamics of Ndc80C.

Our goal is to produce a model with the following specific features, in order to capture the essential dynamics of rod-like biomolecules:

• It must be able to incorporate thermodynamically-correct thermal fluctuations, which are essential to the dynamics of mesoscale biological systems.

• The rods should also be able to undergo bending, torsion and extension.

• Rods should be able to correctly represent 'hinge regions' in biomolecules, and so must also allow for elastic properties that are inhomogeneous and anisotropic.

• The model must allow for a non-straight equilibrium structure. Many biological objects (such as tubulin, which is found in the spindle microtubules) are bent in their equilibrium configuration, and this is integral to their function.

• It should be computationally inexpensive, allowing simulation of the equivalent of millions of atoms for timescales up to seconds.

Although, some important biopolymers, for example actin and microtubules are effectively inextensible, this limit is not straightforward because the inextensibility constraint requires that the random forces must be confined to space in which the system motion is constrained. Algorithms for achieving this have been developed by Grassia and Hinch,[17] Morse[18] and Liverpool.[19]

2 Definition of the rod model

2.1 Kirchoff and Cosserat curves

Almost all models of flexible thin rods describe their configuration in terms of the Frenet triad and Frenet–Serret relations.[20,21] A rod is represented as a continuous material curve. Each point on the curve has three orthonormal vectors associated with it: the tangent vector, the normal vector, and the binormal vector (Fig. 2). A curve with normal and binormal vectors (also called director vectors) is said to be a directed curve.

Most rod models use a formulation by Kirchoff,[13] or by the Cosserat brothers[6] to describe their energies. Elastic rods that are based on these curves are called Kirchoff or Cosserat rods.

2.2 Rod construction and notation

We construct a discretisation of the curve shown in Fig. 2 as a sequence of extensible straight rod sections connecting a set of discrete nodes (Fig. 3).

This discretisation is based on the discretisation from Bergou et al.,[22] but substantial modifications have been made in order to better support modelling of biological molecules: the addition of extension energy, the removal of inertia, the addition of viscous drag, the addition of thermal noise, changes in the bend energy formula to support hinge regions, and changes to the twist energy formula to support arbitrary equilibrium twisting. These differences will be discussed in greater detail in Section 2.4. The nodes are located at positions r_i,
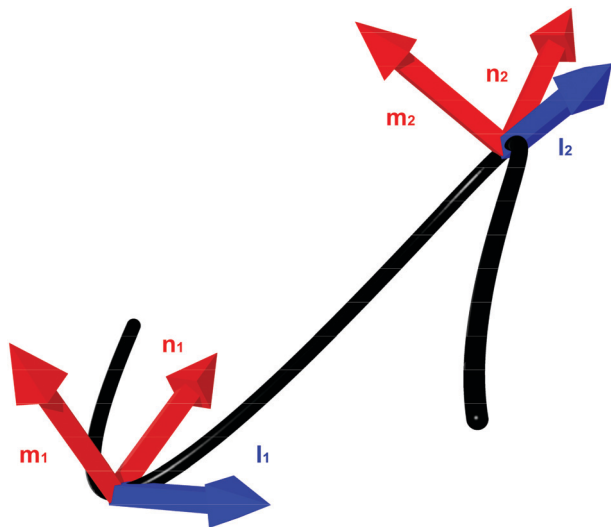
Fig. 2 An example of a continuous framed curve. The tangent, l, and material axes represented by the normal vector m and binormal vector n are shown at two points on the curve.
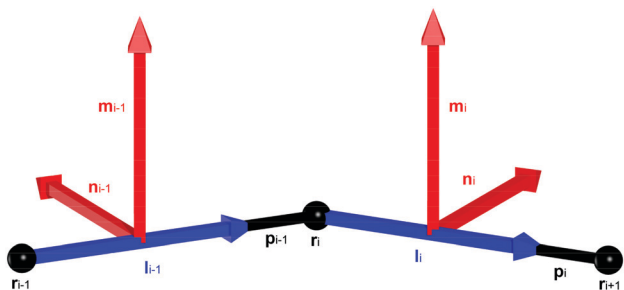

Fig. 3 Discretisation of a continuous framed curve. The curve is constructed from discrete segments p_i that connect together nodes r_i. Each segment has associated material axes (m_i and n_i) and a tangent vector (l_i).

where $i \in \{0, N-1\}$, so that there are $N-1$ segments, with end-to-end vectors $p_i = r_{i+1} - r_i$, and the unit vector along the rod $l_i = \hat{p}_i = p_i/|p_i|$.

To represent the internal twisting of the rod, we define the material axes, m_i and n_i, to be perpendicular to l_i, such that $\|l_i\| = \|m_i\| = \|n_i\| = 1$ and $l_i \cdot m_i = m_i \cdot n_i = n_i \cdot l_i = 0$. The choice of initial direction of m_i and n_i is arbitrary except that they must obey these constraints. However, these vectors are used to represent the relative twisting of rod elements, and form the basis of a local co-ordinate system which rotates as the rod rotates. The local material properties also rotate with the rod, so the directions of m_i and n_i are also used to define the locally anisotropic material properties. A method to initialise a rod which is untwisted at equilibrium is provided in eqn (31).

Finally, to describe rods with an arbitrary equilibrium state, we define equilibrium values $\tilde{p}_i$ and $\widetilde{m}_i$, the configuration from which the rod is deformed.

2.3 Parallel transport

We use the concept of parallel transport to distinguish between changes in the orientation of the material axes arising from the curvature of the centre line and those associated with twist about the centre line. The easiest way to find the twist between two sets of material axes is via the 'parallel transport' of the material axes of one element onto the other.

For two unit vectors a and b we can construct a rotation matrix R that rotates a onto b:

$$R(a, b) = I + [v]_\times + [v]_\times^2 \frac{1}{1+c}, \qquad (1)$$

where

$$v = a \times b \qquad (2)$$

$$c = a \cdot b \qquad (3)$$

and

$$[v]_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}. \qquad (4)$$

To parallel transport the material axes of the ith segment onto the $(i+1)$th segment, we construct the rotation matrix R, which rotates the normalised segment l_i onto the normalised segment l_{i+1}.[23] We then apply that matrix to the material axis m_i to obtain the vector m_i',

$$m_i' = R(l_i, l_{i+1}) \cdot m_i = P(m_i, l_i, l_{i+1}). \qquad (5)$$

The process of parallel transport is illustrated in Fig. 4, where m_i' is the result of the parallel transport of the material axis m_i onto the p_{i+1} th segment. This allows the relative rotations of the material axes for element i and $i+1$ to be compared in a manner that removes the bend between the elements.

2.4 Elastic energy of deformation

We calculate the forces and torques on the rod from the gradients in the elastic energy arising from deforming the rod away from its equilibrium state. This elastic energy is
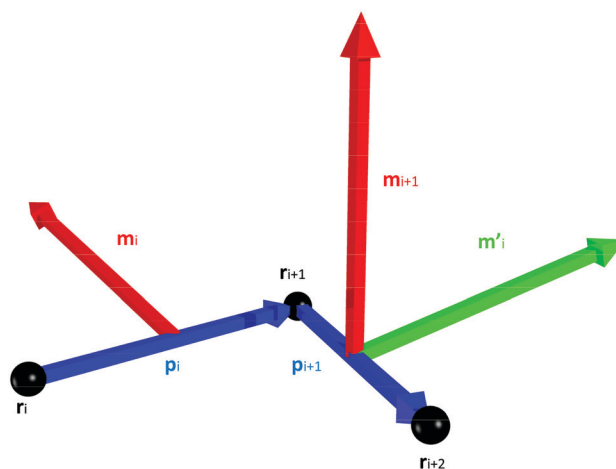

Fig. 4 Parallel transport of the material axis m_i from the p_ith segment to m_i' on the p_{i+1}th segment.

composed of three components – extensional, torsional, and bending.

2.4.1 Extensional deformation. We assume that the extensional deformation is sufficiently small that extensional elastic energy remains in the linear regime, which, for a coiled-coil persists up to an extension to 120% of its original length.[24] Thus, the elastic energy of a single segment due to extension is given by:

$$E_{\text{stretch},i} = \frac{1}{2}k_i(|\boldsymbol{p}_i| - |\tilde{\boldsymbol{p}}_i|)^2, \tag{6}$$

where $|\boldsymbol{p}_i|$ is the length of the ith segment and $|\tilde{\boldsymbol{p}}_i|$ is its equilibrium length (Fig. 5). The value of the spring constant k_i for the ith element can be written as,

$$k_i = \frac{\kappa_{s,i}}{|\tilde{\boldsymbol{p}}_i|}, \tag{7}$$

where $\kappa_{s,i} = YA$, the product of the effective Young's modulus Y and cross-sectional area A. Note that $\kappa_{s,i}$ is therefore a property of local molecular structure, and so is independent of the rod discretisation, whereas the spring constant k_i depends on the discretised element length.

2.4.2 Torsional deformation. To compute the torsional energy, we need to measure the degree of twist from one rod segment to the next, as shown in Fig. 6. We define an angle of rotation, $\Delta\theta_i$, between the material frames of the rod segments i and $i + 1$, such that

$$\Delta\theta_i = \text{arctan2}((\boldsymbol{m}_{i+1} \times \boldsymbol{m}_i') \cdot \boldsymbol{l}_{i+1}, \boldsymbol{m}_i' \cdot \boldsymbol{m}_{i+1}), \tag{8}$$

where $\boldsymbol{m}_{i+1}$ is the material axis of the $(i + 1)$th segment, $\boldsymbol{m}_i'$ is the material axis of the ith segment parallel transported (eqn (5)) onto the $(i + 1)$th segment, and $\boldsymbol{l}_{i+1}$ is the unit vector along the $(i + 1)$th segment. We use the arctan2(y,x) function, which returns the value of $\arctan\left(\frac{y}{x}\right)$, but uses the signs of the parameters to determine which quadrant of the Euclidean plane the angle resides in, thus avoiding a potential sign error.

Within the linear elastic regime, the torsional energy at node i arises from the rotation between the material axes of the adjacent segments:

$$E_{\text{twist},i} = \frac{\beta_i}{L_i}\Big(\text{mod}\Big(\Delta\theta_i - \widetilde{\Delta\theta_i} + \pi, 2\pi\Big) - \pi\Big)^2. \tag{9}$$

Here β_i is the torsion constant (analogous to the stretching constant), $\Delta\theta_i$ and $\widetilde{\Delta\theta_i}$ are the angles between the material
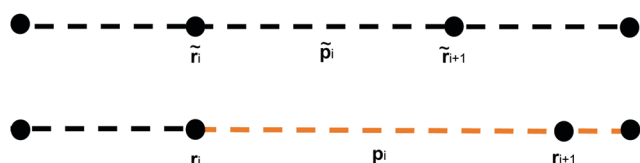

Fig. 5 Changing the length of an element $\boldsymbol{p}_i$ creates a stretching energy E_{stretch}. Note that the stretching energy is a property of an element, and that moving a node results in two elements having different stretching energy (e.g. moving the $\boldsymbol{r}_{i+1}$th node results in changes in the stretching energy from both the $\boldsymbol{p}_i$th and the $\boldsymbol{p}_{i+1}$th node).
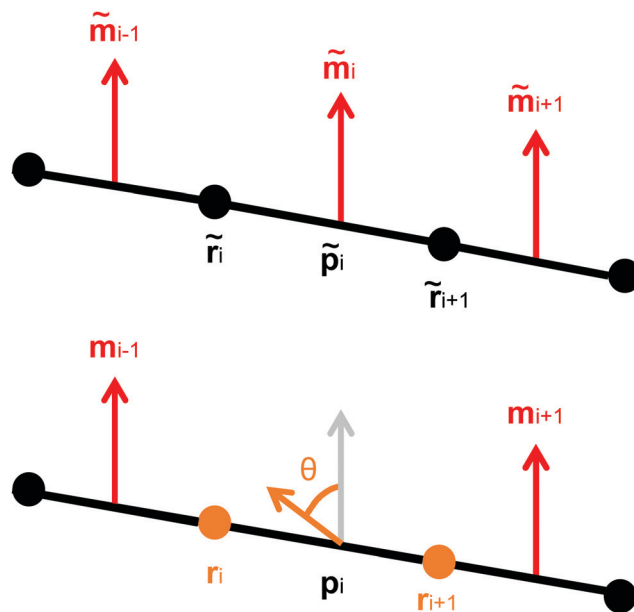

Fig. 6 The energy of torsional deformation is defined between two elements (about a node). Therefore, applying twist to the material axes $\boldsymbol{m}_i$ and $\boldsymbol{n}_i$ will affect the twisting energy about the $\boldsymbol{r}_i$th and $\boldsymbol{r}_{i+1}$th nodes. The rod is normally not straight (as shown here), so parallel transport (Section 2.3) is used to ensure that both material axis vectors are in the same basis.

frames in the current and equilibrium configurations, and

$$L_i = \frac{|\boldsymbol{p}_i| + |\boldsymbol{p}_{i-1}|}{2}. \tag{10}$$

Note that eqn (8) differs from Bergou et al.,[22] who chose to define the equilibrium such that $\widetilde{\Delta\theta} = 0$. The twist energy does not account for the number of turns between two elements being >1, so the discretisation should be chosen so that $\Delta\theta$ is confined to the range $-\pi < \Delta\theta < \pi$. By choosing a periodic function in eqn (9) we ensure that the energy remains continuous.

2.4.3 Bending deformation. To obtain the bending energy (Fig. 7), we compute the curvature binormal, $(k\boldsymbol{b})_i$, which defines the change in orientation between two segments due to bending: $(k\boldsymbol{b})_i$ is orthogonal to both of the segments, with a magnitude equal to twice the tangent of half of the angle between them:

$$(k\boldsymbol{b})_i = \frac{2\boldsymbol{p}_{i-1} \times \boldsymbol{p}_i}{|\boldsymbol{p}_i||\boldsymbol{p}_{i-1}| + \boldsymbol{p}_{i-1} \cdot \boldsymbol{p}_i}, \tag{11}$$

where $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i-1}$ are the ith and $(i - 1)$th segments.

For a rod with an isotropic bending stiffness, the curvature binormal $(k\boldsymbol{b})_i$ is sufficient to calculate the bending energy. However, for a rod with an anisotropic bending stiffness, we need to resolve the components of bend with respect to the local material axes. In our discretisation, the material axes are properties of the elements, whereas the curvature binormal is defined at the node. To deal with this discrepancy, Bergou et al. average the energy arising from projecting the curvature binormal of the node onto the material axis of each the two adjacent rod
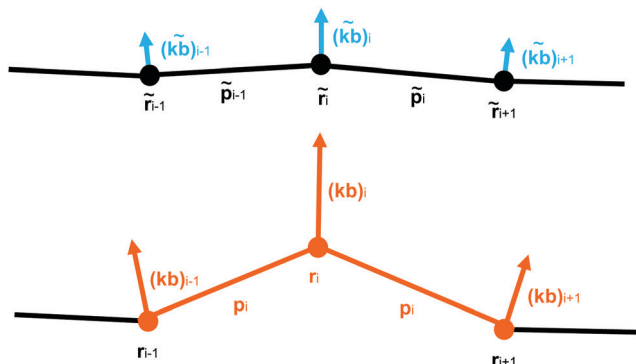
Fig. 7 A bend about a node increases the length of the curvature binormal associated with that node. For an isotropic, untwisted rod, the bending energy is proportional to the square of the curvature binormal. Note that the bending energy is the property of a node, not of an element, and that moving a single node won't just affect the bending energy about that node, but the two nodes on either side of it as well.

elements. However, this has the effect of averaging the anisotropic bending modulus between the material directions of the two elements. Such averaging is undesirable if, for example, there is a local 'hinge' with strongly localised flexibility, as averaging between elements will remove the low modulus in the flexible direction. Therefore, to retain the functionality of a localised hinge, we need to define a material axis at the node. We do this by defining an intermediate segment, centred on this node, from a weighted average of the two segments on either side of this node, called the mutual segment (Fig. 8).

We define the unit vector of the mutual segment as,

$$l^{\mathrm{m}} = \frac{P^{\mathrm{m}}}{|P^{\mathrm{m}}|}, \tag{12}$$

where $P^{\mathrm{m}} = \left(\frac{l_{i-1}}{|p_{i-1}|} + \frac{l_i}{|p_i|} \right)$,

and l_i and l_{i-1} are the unit vectors of the segments on either side of the node. We use an inverse weighting of element lengths in the average so that the orientation of the mutual material frame is more affected by the shorter of the two adjacent elements (whose centre is closer to the node).

Next, we form the mutual material axis at the node by first parallel transporting the material axes of the two adjacent elements onto the mutual segment,
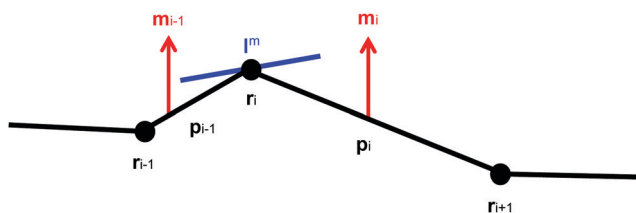
$$m_i^- = P(m_{i-1}, l_{i-1}, l^{\mathrm{m}}), \tag{13}$$

Fig. 8 A mutual rod segment l^{m} about the ith node. The segment is weighted more towards the ith node, which it is closer to.

$$m_i^+ = P(m_i, l_i, l^{\mathrm{m}}), \tag{14}$$

and then by averaging these two material axes, weighted according to inverse element length, and then normalising to give:

$$m^{\mathrm{m}} = \frac{M^{\mathrm{m}}}{|M^{\mathrm{m}}|} \tag{15}$$

where $M^{\mathrm{m}} = \left(\frac{m_i^-}{|p_{i-1}|} + \frac{m_i^+}{|p_i|} \right)$.

By projecting the curvature binormal into the components of the material axes of the new mutual segment, we obtain the material curvature ω, the amount of bending in the local material axis frame,

$$\omega_i^{\mathrm{m}} = ((kb)_i \cdot n^{\mathrm{m}}, -(kb)_i \cdot m^{\mathrm{m}})^{\mathrm{T}}, \tag{16}$$

where $n^{\mathrm{m}} = m^{\mathrm{m}} \times l^{\mathrm{m}}$, and kb_i is the curvature binormal about the ith node.

Assuming that the material properties of the rod are such that torque and curvature are linearly related, the centreline curvature can be used to define the bending energy similarly to the previous energies – as a quadratic potential centred about $\tilde{\omega}_i^{\mathrm{m}}$, the equilibrium material curvature:

$$E_{\mathrm{bend}, i} = \frac{1}{2\tilde{L}_i} \left[\left(\omega_i^{\mathrm{m}} - \tilde{\omega}_i^{\mathrm{m}} \right)^{\mathrm{T}} \cdot B^i \cdot \left(\omega_i^{\mathrm{m}} - \tilde{\omega}_i^{\mathrm{m}} \right) \right], \tag{17}$$

where $\tilde{L}_i$ is given by eqn (10) evaluated for the equilibrium configuration, and B^i is the bending stiffness matrix in the local material axis frame, a positive definite 2×2 matrix.

This formulation does not assume that p_{i-1} and p_i are in similar directions, so it can account for arbitrarily large bend angles about a single element.

2.5 Dynamical equations

The changes to the configuration of the assembly of rod elements is defined by the changes of node positions, Δr_i and rotations $\Delta \theta_i$, which determine the rotations of the material axes orientations (m_i). The translational motion of each node is given by a stochastic equation of the form

$$\Delta r_i = \mathscr{M}_i \cdot (F_i + f_i) \Delta t, \tag{18}$$

where Δt is the simulation timestep, $\mathscr{M}_i$ is the mobility tensor of the rod segment in the fluid medium, F_i is the internal elastic force, and f_i is the random force from stochastic thermal noise. This equation assumes that the system is overdamped, so that the inertia of the rod is negligible. We also neglect hydrodynamic interactions between the rod elements, but include a resistive (damping) force for the motion of an isolated rod element against a background fluid medium. This is therefore a Brownian equation of motion. For a general axisymmetric body the mobility tensor will be anisotropic. However, for simplicity we have approximated the mobility as being isotropic and equal to mobility of a sphere of a radius, a_i equal to half the equilibrium length of each segment:[25]

$$\mathscr{M}_i = \zeta_i^{-1} \mathbf{I}, \tag{19}$$

where $\zeta_i = 6\pi\mu a_i$ and μ is the dynamic viscosity of the medium.

Similarly, for the rotational motion of the material axes,

$$\Delta\theta_i = \frac{\Delta t}{\zeta_\theta}(\tau_i + g_i), \tag{20}$$

where g_i is a random thermal torque, τ_i is the torque due to the internal elastic forces (23), and ζ_θ is the corresponding friction constant for rotation. For the rotational friction constant we use the rotational drag on cylinder of radius a and length $|\underline{p}_i|$:

$$\zeta_\theta = 8\pi\mu a^2 \cdot |\underline{p}_i|. \tag{21}$$

For a particular configuration of nodes and material axes, the forces acting on nodes (F_i) and the torques acting on material axes (τ_i) can be determined from the partial gradients of the energy:

$$F_i = \frac{\partial E}{\partial r_i}, \tag{22}$$

where F_i is the force on the ith node at r_i due to the energy gradient, and

$$\tau_i = \frac{\partial E}{\partial \theta_i}l_i, \tag{23}$$

where τ_i is the torque resulting from the energy gradient of segment i, l_i is the unit vector representing the axis of rotation, and the total internal energy, $E = E_{stretch} + E_{twist} + E_{bend}$, where $E_{stretch}$, E_{twist} and E_{bend} are obtained by summing the node and element contributions to stretch, twist and bend energies from eqn (6), (9) and (17).

Due to the complexity of these formulas we compute the gradient numerically using central differences,

$$\frac{\partial E}{\partial x} \simeq \frac{\Delta E}{\Delta x} = \frac{E(x + \Delta x) - E(x - \Delta x)}{2\Delta x}, \tag{24}$$

where x is some co-ordinate (position or angle) in the system. When a node is translated, or when a material axis is twisted, it changes the energies of the two nodes on either side of it. Fig. 5–7 display the affected geometry in orange. Therefore, computing the gradient in energy at node i requires information about a 5-node window of the rod between r_{i-2} and r_{i+2}.

When a node is translated, the elements and the material axes on either side of that node change. The element is updated according to the discretisation given in Section 2.2. The material axis is parallel transported (Section 2.3) from the old rod direction to the new one. More details are available in Section 1 of the ESI.†

Using the drag given by eqn (19), the force acting on each node from thermal noise is given by the fluctuation dissipation theorem,[26]

$$f_i = \sqrt{\frac{24k_B T\zeta_i}{\Delta t}}R, \tag{25}$$

where T is the temperature of the system, ζ the friction constant, Δt the timestep, k_B is Boltzmann's constant, and R is a random vector, where R_x, R_y and R_z are independently drawn from uniform distributions in the range $-0.5 \leq R_{x,y,z} \leq 0.5$, such that $\langle R \rangle = 0$ and $\langle R_i R_j \rangle = \frac{1}{12}\delta_{ij}$.

It would be possible to include hydrodynamic interactions by replacing eqn (18) for each segment with a single equation for the entire rod with a single mobility matrix including the hydrodynamic coupling between the elements. However, this would result in replacing ζ_i in eqn (25) with the inverse of the mobility tensor, which is computationally more expensive. If the rod remains nearly straight that the effect of hydrodynamic interactions can be approximated through a difference in the local mobility parallel and perpendicular to the rod axis.

The rotational thermal torque is given by

$$g_i = \sqrt{\frac{24k_B T\zeta_\theta}{\Delta t}}R \tag{26}$$

where T is the temperature of the system, ζ_θ is the torsional friction, Δt is the timestep, k_B is Boltzmann's constant, and R is a uniform random variable in the range $-0.5 \leq R \leq 0.5$.

To apply the rotation $\Delta\theta$ to the material axis, we use Rodrigues rotation formula:

$$v_{rot} = v \cos\Delta\theta + (k \times v)\sin\Delta\theta + k(k \cdot v)(1 - \cos\Delta\theta) \tag{27}$$

where v_{rot} is the resultant vector, $\Delta\theta$ is the angle to rotate, v is the original vector and k is the axis of rotation. In this case, we would be rotating the vector m_i about the axis l_i.

For information regarding the structure, implementation and performance of this algorithm, see Section 2 of the ESI.†

3 Parameterisation of rods from atomistic molecular dynamics

In this section, we apply the general model for rod-like biomolecules described in Section 2 to the specific case of the Ndc80 protein complex. Ndc80C is a long, flexible molecule comprised of two coiled-coil sections joined by an unstructured loop. As a consequence, this molecule has both a non-straight equilibrium shape and an inhomogeneous, anisotropic bending modulus, and so allows us to test all of the features of this model. A cartoon depiction of the Ndc80C structure is shown in Fig. 9, showing the connectivity of Ndc80C's four sub-units (Ndc80 in blue, Spc24 in purple, Nuf2 in yellow and Spc25 in green) along with the positions and residue numbers of the cross-links.

First, we will construct an all-atom representation of Ndc80C. Then, we will extract coarse grained rod model parameters from the dynamics of a simulation of the all-atom Ndc80C. Finally, we will compare the results from these models to each other and to measurements acquired from negative stain EM images.[27]

3.1 Creation of an all-atom Ndc80C model

All-atom molecular dynamics software packages such as AMBER[28] and GROMACS[29] are commonly used to study the dynamics of biological molecules.[30] Unlike KOBRA, these
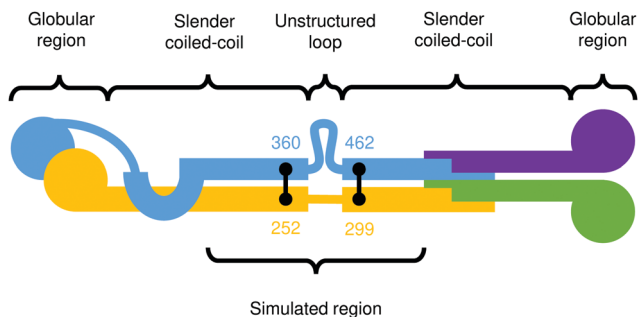
Fig. 9 Cartoon depiction of the Ndc80C protein complex.[3] The colours distinguish the four connected proteins in the complex, and the black lines denote cross-links at their respective residue numbers.

simulations consider the position and connectivity of every atom (and hence, every residue) contained within a given molecule. The dynamics are computed using a force field, from pre-computed potentials that describe every atom and bond in the system. Therefore, to run an MD simulation of Ndc80C, we need to construct a model of the structure by combining information from experimental studies.

As a starting point, we use the so-called 'Bonsai' Ndc80C molecule[3] (2VE7). This molecule contains both globular domains and heavily-truncated coiled-coil region 85 Å in length.

Ciferri *et al.*[3] provide the positions of cross-links between the two coiled-coils and a prediction of the entire secondary structure by type: a long coiled-coil with a small, unstructured loop at the centre. They also indicate the relative residue numbers at which these features occur as shown in Fig. 9.

To construct an all-atom model of Ndc80C, we use the spatial parameters (such as the pitch and coiled-coil radius) extracted from the Bonsai Ndc80C combined with the residue sequence from the full-length molecule. We remove the globular regions at both ends and retain the central coiled-coil region. Details of the sequences are available in the ESI.† We use ISAMBARD, a software package designed for modelling, analysis and parametric design of proteins,[31] to extract the coiled-coil parameters – such as pitch, radius, crick angles, and ϕC_{α} values – from the coiled-coil present in the Bonsai molecule. The radius was found to be 7.62 ± 0.12 Å, the pitch 203 ± 26 Å. ISAMBARD is then used to create a coiled-coil backbone with these parameters out of the full Ndc80C residue sequence. This structure has no side-chains, so we recreate the side chains from the residue sequence using SCWRL4, a protein side chain predictor.[32]

We use the QUARK *ab inito* protein structure prediction server[33] to create predictions of the structures in loop region from the residue sequence. These structures were peptide-bonded together with the two coiled-coils using UCSF Chimera.[34] The overall length of the generated Ndc80C is 48 nm.

Finally, we refine the resulting model using ModRefiner,[35] and minimise it using the YASARA minimisation server.[36] This brings the protein, which has been created with somewhat arbitrary bond angles, closer to its native state. We then run simulations of this structure using AMBER[28] with an implicit (GBSA) solvent, for 50 nanoseconds. The simulation trajectories and input files are available in the ESI.†

3.2 Model building

To construct a rod model for Ndc80C, we need to obtain suitable values for the physical parameters κ_{s}, β and $\boldsymbol{B}$ for each node and element. These parameters are obtained from Ndc80C's dynamics – we run an atomistic simulation of Ndc80C and aim to select rod parameters that reproduce the same local fluctuations in shape.

We first map the all-atom MD trajectory directly onto a coarse-grained trajectory for an equivalent rod model. We define the position of nodes in this rod model by averaging over the positions of small clusters of atoms:

$$\boldsymbol{r}_i^{\mathrm{r}} = \frac{1}{k_{\max} - k_{\min}} \sum_{k=k_{\min}}^{k_{\max}} \boldsymbol{r}_k^{\mathrm{a}}, \qquad (28)$$

where $\boldsymbol{r}_i^{\mathrm{r}}$ is a rod node at index i, $k_{\min}$ and $k_{\max}$ are the atom indices denoting the edges of the ith cluster, and $\boldsymbol{r}_k^{\mathrm{a}}$ is the position of the atom at index k. Using only small clusters means that the rod configuration is not unnaturally smoothed during coarse-graining. The all-atom Ndc80C model was coarse-grained to 14 nodes from 2668 atoms. The optimal length for rod segments depends on the system being discretised. It should be as coarse as possible, while being fine enough to capture information about the implicit shape and dynamics of the molecule being modelled. In the case of Ndc80C, the rod should be fine enough to resolve the molecular hinge, but each element should be no shorter than a turn of the coiled-coil, as at this resolution, it would be necessary to represent the coiled-coil explicitly using two rods.

Each node was averaged from a 10-atom wide cluster according to eqn (28) (see Fig. 10). This number is also system-dependent, for Ndc80C it was chosen such that there is more than a factor of 10 difference between number of atoms per rod element and the number of atoms being used to set a node position.

Having defined the nodes, we now need to define the material axis of each coarse-grained segment in order to measure twist. One method for doing this is to first calculate the vector $\boldsymbol{q}_i$ between the opposite atoms in the two chains of the coiled-coil (Fig. 11):

$$\boldsymbol{q}_j = \frac{\boldsymbol{r}_j^{\mathrm{A}} - \boldsymbol{r}_j^{\mathrm{B}}}{\left| \boldsymbol{r}_j^{\mathrm{A}} - \boldsymbol{r}_j^{\mathrm{B}} \right|}, \qquad (29)$$
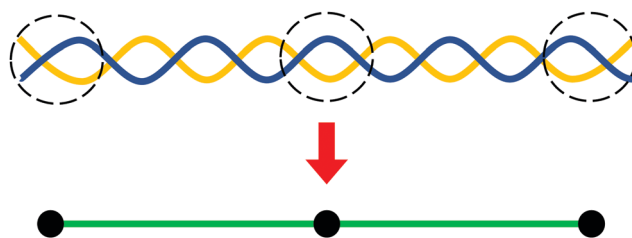

Fig. 10 To coarse-grain the atomistic Ndc80C trajectory, the co-ordinates of groups of atoms are averaged according to clusters of atom indices.
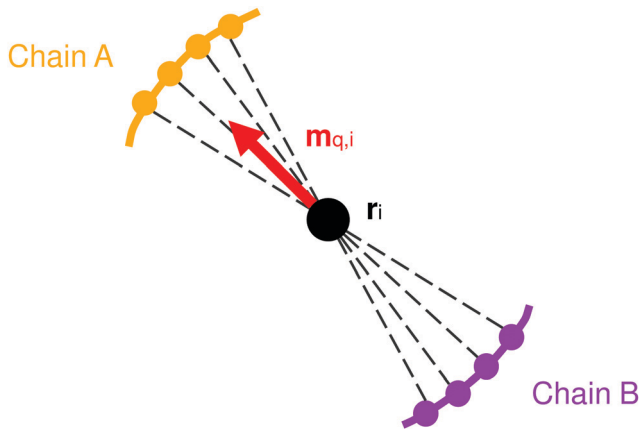
Fig. 11 Schematic view of the coiled-coil cross-section, showing how the material axis is computed. The dashed lines are used to compute $\boldsymbol{q}_i$ for each atom, which are then averaged to $\boldsymbol{m}_{q',i}$. For clarity, only four atoms in each chain are depicted.

where $\boldsymbol{r}_j^{\mathrm{A}}$ is the position of the *j*th atom in chain A, and $\boldsymbol{r}_j^{\mathrm{B}}$ is the position of the atom in chain B which is closest to $\boldsymbol{r}_j^{\mathrm{A}}$ at equilibrium. In practice, for each rod we select a 10 atom pairs which are located halfway between the atoms defining two nodes at the end of the rod; we take the average $\boldsymbol{q}_j$ from those atom pairs, as indicated in Fig. 11. A material axis $\boldsymbol{m}_{q,j}$ can then be defined from the orthonormal projection of this vector from the normalised element axis, $\boldsymbol{l}_j$,

$$\boldsymbol{m}_{q,j} = \frac{\boldsymbol{q}_j - (\boldsymbol{q}_j \cdot \boldsymbol{l}_j)\boldsymbol{l}_j}{|\boldsymbol{q}_j - (\boldsymbol{q}_j \cdot \boldsymbol{l}_j)\boldsymbol{l}_j|}. \quad (30)$$

However, the material axes of adjacent elements obtained from eqn (30) may point in quite different directions, and although the above equations for calculating deformation energy permit this, it is practically better (*e.g.* when defining the bending stiffness matrix) to maintain similar directions for the material axes of adjacent elements. The material axes of two adjacent elements can be considered to be in the 'same' direction if the material axis $\boldsymbol{m}_i$ of element *i*, when parallel transported onto element $i + 1$, points in the direction of the material axis $\boldsymbol{m}_{i+1}$ of that element.

Consequently, we use the following procedure for calculation of material axes. We begin in the equilibrium configuration of the atomistic model. In this equilibrium configuration, we obtain the material axis $\boldsymbol{m}_0$ of the first element according to eqn (30). We then generate the material axis for all other rod elements in the equilibrium configuration by parallel transport, by iterating along the rod:

$$\widetilde{\boldsymbol{m}_i} = P\left(\tilde{\boldsymbol{m}}_{i-1}, \frac{\widetilde{\boldsymbol{p}_{i-1}}}{|\widetilde{\boldsymbol{p}_{i-1}}|}, \frac{\widetilde{\boldsymbol{p}_i}}{|\widetilde{\boldsymbol{p}_i}|} \right). \quad (31)$$

For each element *i* we can also calculate a candidate material axis $\boldsymbol{m}_{q,i}$ from the local atomistic positions according to eqn (30). In the equilibrium configuration we may compare the material axis $\boldsymbol{m}_i$ obtained by parallel transport with the material axis $\boldsymbol{m}_{q,i}$. These two vectors will in general differ by some rotation angle
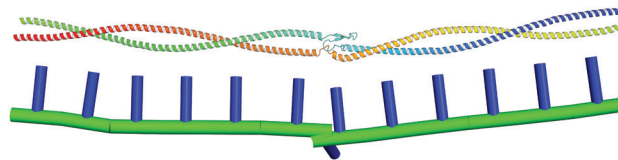

Fig. 12 Top: Completed atomistic Ndc80C model after minimisation. Bottom: Simplified rod model showing equilibrium atomistic structure. The blue lines represent the equilibrium orientation of the material axes, and their twisting indicates the coiled-coil pitch. The radius display is arbitrary.

$\widetilde{\Delta\theta_i^{\mathrm{u}}}$, which we then store. Then, during the subsequent dynamics of the MD trajectory, at any moment we can generate the instantaneous $\boldsymbol{m}_{q,i}$ according to eqn (30), and calculate the instantaneous material axis $\boldsymbol{m}_i$ by rotating $\boldsymbol{m}_{q,i}$ about the rod by signed angle $\widetilde{\Delta\theta_i^{\mathrm{u}}}$, using the Rodrigues rotation formula (eqn (27)).

The Python script that builds this model is available in the ESI,† and as part of the FFEA software package. The model, as built, is illustrated in Fig. 12.

3.3 Model parameterisation

Having mapped the MD simulations onto a trajectory for an equivalent rod model, our strategy for model parameterisation is to select parameters such that the local mean square fluctuations in bending, twisting and extension match those observed in the equivalent rod model from the MD trajectory. From this trajectory, we therefore compute (for each rod, or node, *i*) the mean square fluctuations in rod length $\langle (|\boldsymbol{p}_i| - |\tilde{\boldsymbol{p}}_i|)^2 \rangle$, and in twisting angle $\langle \Delta\theta_i^2 \rangle$ where $\Delta\theta_i$ is the difference between instantaneous twisting angle and its time average value. Since bending is parameterised by a two-component vector $\boldsymbol{\omega}_i^{\mathrm{m}}$ (see eqn (16)), fluctuations in bending are parameterised by a covariance matrix for bending fluctuations, for node *i*:

$$\boldsymbol{C}_{i,\alpha\beta} = \langle \Delta\omega_{i,\alpha}^{\mathrm{m}} \Delta\omega_{i,\beta}^{\mathrm{m}} \rangle, \quad (32)$$

where $\Delta\omega_{i,\alpha}^{\mathrm{m}}$ is the difference between α component of the bending vector $\boldsymbol{\omega}_i^{\mathrm{m}}$ and its time average value (where $\alpha = 1$ or 2).

We aim to parameterise the dynamical rod model so as to match the above mean square fluctuations. As an initial estimate for the model parameters, we note that the energies defined in eqn (6), (9) and (17) are all quadratic in the respective parameters. We might expect fluctuations to be distributed normally and the equipartition theorem to apply. This produces initial estimates for the parameters as:

$$k_i = \frac{k_{\mathrm{B}}T}{\left\langle (|\boldsymbol{p}_i| - |\tilde{\boldsymbol{p}}_i|)^2 \right\rangle}, \quad (33)$$

$$\beta_i = \frac{L_i k_{\mathrm{B}} T}{2 \langle \Delta\theta_i^2 \rangle}, \quad (34)$$

$$\boldsymbol{B}^i = k_{\mathrm{B}} T L_i \cdot \boldsymbol{C}_i^{-1}. \quad (35)$$

However, the above results neglect the three dimensional nature of the rod model dynamics, and the non-linear

geometrical effect this has on the probability distributions of rod model variables. A uniform density of states for the x-, y- and z-coordinates of node positions does not equate to a uniform density of states for rod length and bending co-ordinates. As a result, the probability distributions for these variables are typically perturbed from their initially expected normal distributions (one can think of this as an extra entropic contribution to the free energy of fluctuations arising from the density of states in the space of the bending or rod length co-ordinates).

We have found this problem to be especially prominent for the bending fluctuations. If we use eqn (35) for the bending stiffness matrix, then typically the mean square fluctuations in the resulting rod model do not match those expected. However, they are sufficiently close that an iterative scheme is able to rapidly converge on the correct parameterisation. For a given node i, suppose we are aiming to reproduce bending fluctuations $\boldsymbol{C}_{\text{target}}$ as measured from the all atom MD simulations. Running a KOBRA model built using the bending energy matrix $\boldsymbol{B}_{\text{old}}$, given by application of eqn (35), will result in a trajectory with observed fluctuations $\boldsymbol{C}_{\text{old}}$ which will in general be different from $\boldsymbol{C}_{\text{target}}$ (even after reducing statistical errors by running a long trajectory). By considering the expected change in $\boldsymbol{C}$ from a small change in $\boldsymbol{B}$, we can obtain a new guess for an optimal bending energy matrix as:

$$\boldsymbol{B}_{\text{new}} = k_{\text{B}}TL_i(\boldsymbol{C}_{\text{target}} - \boldsymbol{C}_{\text{old}} + k_{\text{B}}TL_i\boldsymbol{B}_{\text{old}}^{-1})^{-1}. \quad (36)$$

A KOBRA simulation run using this new guess at the bending energy matrix will result in bending fluctuations closer to the target value. If necessary, the iteration can be repeated, but we have found that a single iteration is usually sufficient to satisfactorily reproduce the target bending fluctuations within typical additional error produced from statistical sampling of trajectories.

Validation of equipartition of energy, and extraction of bending parameters from dynamical simulations, are described in more detail in Sections 3 and 4 of the ESI.†

4 Results

4.1 All-atom parameterisation results

Using the method described in Section 3.3, and the model from Section 3.1, we can determine a set of rod parameters for Ndc80C. The unstructured loop acts as a flexible hinge linking together stiffer coiled coil regions of the molecule. The average values of these constants for the entire molecule (including the flexible hinge) are given here in Table 1.

Table 1 Average values of rod parameters for Ndc80C. $\boldsymbol{B}$ is assumed to be isotropic in this case, so the values quoted are the diagonal elements of the $\boldsymbol{B}$ matrix

κ	$(1.718 \pm 0.017) \times 10^{-11}$ N m^{-1}
$\boldsymbol{B}$	$(5.15 \pm 0.29) \times 10^{-31}$ m^4 Pa
β	$(1.32 \pm 0.07) \times 10^{-29}$ nm^2
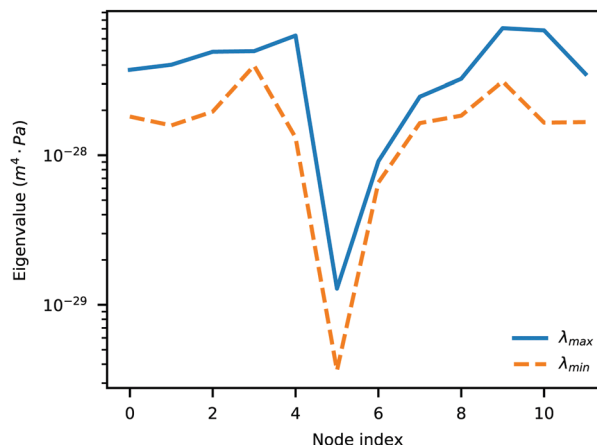
Fig. 13 The eigenvalues of the bending energy matrix $\boldsymbol{B}^i$ for the rod calculated from the atomistic Ndc80C trajectory. Here, the two lines represent the maximum and minimum eigenvalues, not any particular axis.

In addition to making it more flexible, Ndc80C's hinge region means that it is slightly more susceptible to bending in one axis than in the other. This hinge is localised to a small region in the centre of Ndc80C, so we need to examine the parameters extracted on a per-node basis.

Fig. 13 indicates the two eigenvalues of $\boldsymbol{B}^i$ as a function of the node number. It shows a localised region of decreased stiffness between nodes 4 and 6 in the parameterised rod model, which corresponds to the unstructured loop region (the hinge) in the atomistic model. This hinge is also observed in the negative stain EM imaging of Ndc80C,[27] which will be examined in more detail in Section 4.2.1.

4.2 Comparison of dynamics with atomistic molecular dynamics and experimental data

The parameterisation from Section 4.1 was used to create a KOBRA rod model and run a simulation comprised of 2×10^7 steps, with a timestep of 10 ps, for a total simulation time of 200 µs. Note that this is more than three orders of magnitude longer than the atomistic simulations.

4.2.1 Comparison of molecular kink angles. Wang *et al.*[27] use negative stain EM to obtain 2D images of a variety of Ndc80C conformers. They force the Ndc80C molecules to lie on a flat carbon support, and measure the 'kink angle', defined as the angle between the two halves of the molecule, before and after the unstructured loop. This provides a distribution of angles, based on 83 observations, and so gives an experimental measurement the flexibility of Ndc80C.

From both the KOBRA rod and MD atomistic trajectories, the kink angles can be measured using the end-to-end vectors $\boldsymbol{p}_{\text{a}}$ and $\boldsymbol{p}_{\text{b}}$ of the two halves of the coiled-coil region (*i.e.* from one end of the coiled-coil to the loop/hinge region, and then from the loop/hinge region to the far end of the coiled-coil):

$$\theta = \arccos\left(\frac{\boldsymbol{p}_{\text{a}} \cdot \boldsymbol{p}_{\text{b}}}{|\boldsymbol{p}_{\text{a}}||\boldsymbol{p}_{\text{b}}|}\right). \quad (37)$$
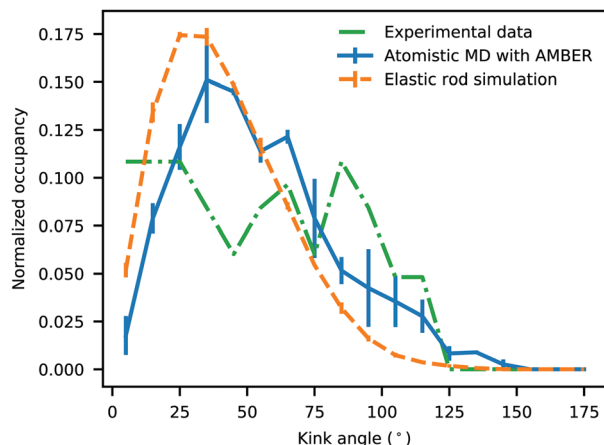
Fig. 14 Kink angles for experimental data, rod simulation and atomistic simulation. Angles are given as absolute values.

This angle is computed for each frame in the trajectory, then binned in increments of 10 degrees in order to preserve parity with the experimental data, and the resulting distribution normalised (see Fig. 14).

The distribution of kink angles for the KOBRA rod and atomistic trajectories are very similar. Both possess a broad maximum of bend angles between 25 and 75 degrees centered on 50 degrees, that falls off rapidly for angles above 100 degrees. This occurs because of the density of states from projection of the two possible bending directions onto a single bend angle. Since the distribution of bend angles at each node from the MD trajectory was used to parameterise the KOBRA rod simulation this can be interpreted as an indication of a successful parameterisation.

The distribution of the experimental angles retains the same general features. The experimental distribution does not replicate the maximum at 50 degrees. However, this discrepancy arises in part because the experimental resolution was not able to resolve bend angles less than 30 degrees – so all angles in this range were inserted into a single bin. Otherwise, the experimental data broadly replicates the features of the distributions found from simulation although the range of the bend angles is slightly broader. It should also be noted that the experimental distribution is constructed from only 83 samples, compared to 10 000 frames for the rod trajectory and 50 000 frames for the atomistic trajectory.

The rod simulation trajectories and input files are available in the ESI.†

4.2.2 Comparison of principle components. To understand which types of motion are conserved between the all-atom simulation and the KOBRA simulation, we can analyse the trajectories from the two simulations using principal component analysis (PCA).[37] PCA is a statistical method used to reduce the dimensionality of molecular dynamics data sets, by identifying the principal eigenmodes of motion (the principle components) resulting from thermal (or other) fluctuations. Each eigenmode represents a different structural deformation. The principal components are ordered from the largest to the

smallest value of the associated eigenvalue (and thus, amplitude of the component). PCA was performed using the software package pyPcazip.[38]

Fig. 15 shows the dot product matrix comparing the rod and all-atom simulations. Each cell shows the dot product of the eigenvector of the rod mode with the corresponding all-atom mode. A dot product close to one means that the eigenvectors are highly correlated. The eigenvectors are sorted by eigenvalue size, so if the matrix is diagonal, it also means that the relative magnitude of the various modes of bending are in the same order. The highest correlation is between the first modes, which correspond to bending about the hinge. Although diagonal elements of the matrix dominate, there are some significant off-diagonal components, which could be evidence of mode mixture, particularly between mode 2 and 4. For reference, when assessing correlations between eigenmodes from multiple all-atom trajectories, these normally display a significantly smaller degree of diagonal correlation between PCA eigenmodes from separate runs on the same model[5] than is shown in Fig. 15.

To observe this mode mixture, and visualise the motion represented by the first few PCA modes, we can create 'PCA animations', that show the range of motion corresponding to each eigenmode. The PCA animations are created using the following formula:

$$r_i^{\text{anim}} = r_i + (\lambda f)v, \qquad (38)$$

where r_i^{anim} is the animated node, r_i is the original node, λ is the eigenvalue for that node, v the eigenvector, and f is a scalar between 0 and 1. The resulting figure shows how the nodes oscillate within that principal component.

Fig. 16 compares the range of motion exhibited in the first four principal component eigenmodes obtained from the rod and the all-atom MD simulations. The first mode shows a central hinge, with similar magnitudes of bending for the rod and atomistic model. The second and third all-atom modes show the fluctuation of two different coiled-coil regions. While the second and third principal components initially look dissimilar,
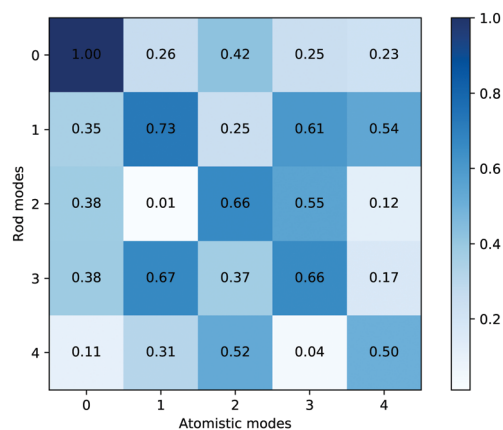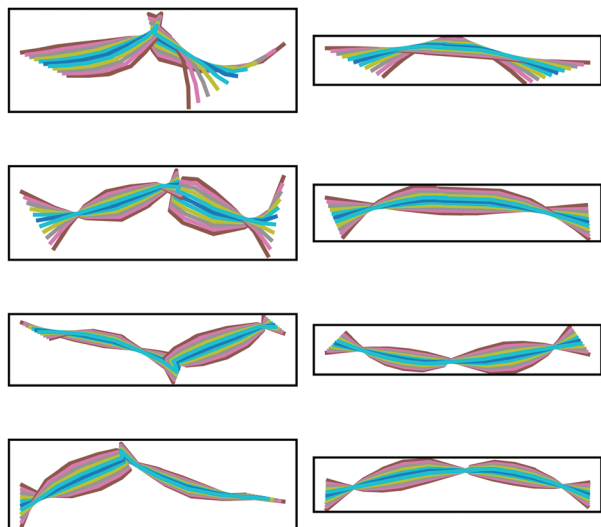

Fig. 15 Dot product matrix of the average principal component eigenvectors. The values shown are normalised to the size of the largest dot product.

Fig. 16 Comparison between the principal components of both trajectories. Left: Original, all-atom trajectory, coarsened to 14 nodes. Right: Rod simulation. Both models were into a plane containing the end-to-end vectors of two halves of the molecule. The principal components appear in decreasing order of size.

the dot product matrix suggests there is mixing between these modes. For example, modes 1 and 2 of the atomistic trajectory show motion on opposite sides of the hinge, whereas in the corresponding rod nodes, this motion is distributed evenly between both modes.

Together, Fig. 15 and 16 indicate a strong correlation between the motions involved in the atomistic MD simulation and the KOBRA rod trajectories. The degree of correlation is typical (and in fact good) in comparison to PCA mode correlations between similar MD simulations, or between MD and coarse grained simulations. One reason for the discrepancy in correlation is the length of the MD trajectories, which are necessarily short (due to computational constraints), and therefore do not sample a large number of different global configurations (the large scale modes) of the molecule, giving rise to a statistical sampling error. It should be noted that in deriving parameter values for the coarse grained rod model we have used local information, such as the variation in local bend and twist angles, rather than the variation in global configurations. For short MD trajectories, matching the local bend and twist variation does not necessarily translate to matching the (statistically limited) variations observed in global configurations.

Animations of each normal mode and pcz-format trajectories are available in the ESI† provided with this paper.

5 Conclusion

The KOBRA algorithm provides an efficient simulation technique for elongated biomolecules that produces dynamical rod trajectories and conformers with bend angles that closely resemble all-atom MD and experimental data. While here we have focused only on Ndc80C, KOBRA is designed to be a general model for slender biological objects such as coiled-coils and

alpha helices. The current version provides a robust coarse-graining methodology designed to capture the dynamics of slender biological objects, which can have anisotropic bending moduli, inhomogeneous bending, stretch and twisting parameters, and arbitrary equilibrium configurations. In addition, it provides tools for model building and a large number of tests for validity. KOBRA is also performant enough to handle simulations of extremely large systems (of order (100 nm) over long timescales (of order 1 ms)), such as the kinetochore.

In the future, we will couple the rod-like KOBRA objects with three-dimensional FFEA objects, in order to construct biomolecules comprised of both globular and filamentary structures. In particular, this will enable us to construct a model of Ndc80C featuring the globular domains at each end, and potentially larger systems including more kinetochore proteins. We will also intend to implement long range hydrodynamics.

The software implementing the KOBRA algorithm is available as part of the FFEA software package (ffea.bitbucket.io), and is free to use and modify under the conditions off the GNU GPLv3 software licence. This includes an implementation of the algorithm, the tools used to convert atomistic trajectories to rod structures, parameter extraction, all of the analysis tools, and a simple visualiser. Developer documentation is also available for those who want to add features or implement KOBRA in their own software.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

Notes and references

1 S. Gonen, B. Akiyoshi, M. G. Iadanza, D. Shi, N. Duggan, S. Biggins and T. Gonen, *Nat. Struct. Mol. Biol.*, 2012, **19**, 925–929.

2 A. Suzuki, B. L. Badger, J. Haase, T. Ohashi, H. P. Erickson, E. D. Salmon and K. Bloom, *Nat. Cell Biol.*, 2016, **18**, 382–392.

3 C. Ciferri, S. Pasqualato, E. Screpanti, G. Varetti, S. Santaguida, G. Dos Reis, A. Maiolica, J. Polka, J. G. De Luca and P. De Wulf, *et al.*, *Cell*, 2008, **133**, 427–439.

4 S. Kmiecik, D. Gront, M. Kolinski, L. Wieteska, A. E. Dawid and A. Kolinski, *Chem. Rev.*, 2016, **116**, 7898–7936.

5 A. Solernou, B. S. Hanson, R. A. Richardson, R. Welch, D. J. Read, O. G. Harlen and S. A. Harris, *PLoS Comput. Biol.*, 2018, **14**, e1005897.

6 E. E. Cosserat and F. F. Cosserat, info:edu-jhu-library/collection/brittlebooks/id/barcode/31151000327233, 1909.

7 J.-S. Wang, Y.-H. Cui, T. Shimada, H.-P. Wu and T. Kitamura, *Appl. Phys. Lett.*, 2014, **105**, 043702.

8 S. Neukirch, A. Goriely and A. C. Hausrath, *Phys. Rev. Lett.*, 2008, **100**, 038105.

9 K. Linka and M. Itskov, *J. Mech. Behav. Biomed. Mater.*, 2016, **58**, 163–172.

10 A. Gorielyn, A. Hausrath and S. Neukirch, *Biophys. Rev. Lett.*, 2008, **3**, 77–101.

11 M. L. Smith and T. J. Healey, *Int. J. Non Linear Mech.*, 2008, **43**, 1020–1028.

12 R. S. Manning, J. H. Maddocks and J. D. Kahn, *J. Chem. Phys.*, 1996, **105**, 5626–5646.

13 G. Kirchoff, *J. Reine Angew. Math.*, 2009, **1859**, 285–313.

14 I. Tobias, B. D. Coleman and M. Lembo, *J. Chem. Phys.*, 1996, **105**, 2517–2526.

15 C. Prior, *J. Elasticity*, 2014, **117**, 231–277.

16 D. K. Pai, *Comput. Graph. Forum*, 2002, **21**, 347–352.

17 P. Grassia and E. J. Hinch, *J. Fluid Mech.*, 1996, **308**, 255–288.

18 D. C. Morse, *Advances in Chemical Physics*, John Wiley & Sons, Ltd, 2004, pp. 65–189.

19 T. B. Liverpool, *Phys. Rev. E: Stat., Nonlinear, Soft Matter Phys.*, 2005, **72**, 021805.

20 F. Frenet, *J. Math. Pure. Appl.*, 1852, 437–447.

21 J.-A. Serret, *J. Math. Pure. Appl.*, 1851, 193–207.

22 M. Bergou, M. Wardetzky, S. Robinson, B. Audoly and E. Grinspun, *ACM transactions on graphics (TOG)*, 2008, p. 63.

23 J. v. d. Berg, https://math.stackexchange.com/users/91768/jur-van-denberg, Calculate Rotation Matrix to align Vector A to Vector B in 3d?.

24 A. Zhmurov, O. Kononova, R. I. Litvinov, R. I. Dima, V. Barsegov and J. W. Weisel, *J. Am. Chem. Soc.*, 2012, **134**, 20396–20402.

25 P. Saffman, *J. Fluid Mech.*, 1976, **73**, 593–602.

26 H. Nyquist, *Phys. Rev.*, 1928, **32**, 110–113.

27 H.-W. Wang, S. Long, C. Ciferri, S. Westermann, D. Drubin, G. Barnes and E. Nogales, *J. Mol. Biol.*, 2008, **383**, 894–903.

28 D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham, S. DeBolt, D. Ferguson, G. Seibel and P. Kollman, *Comput. Phys. Commun.*, 1995, **91**, 1–41.

29 M. J. Abraham, T. Murtola, R. Schulz, S. Pall, J. C. Smith, B. Hess and E. Lindahl, *SoftwareX*, 2015, **1–2**, 19–25.

30 D. J. Huggins, P. C. Biggin, M. A. Dämgen, J. W. Essex, S. A. Harris, R. H. Henchman, S. Khalid, A. Kuzmanic, C. A. Laughton, J. Michel, A. J. Mulholland, E. Rosta, M. S. P. Sansom and M. W. v. d. Kamp, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2019, **9**, e1393.

31 C. W. Wood, J. W. Heal, A. R. Thomson, G. J. Bartlett, A. Ibarra, R. L. Brady, R. B. Sessions and D. N. Woolfson, *Bioinformatics*, 2017, **33**, 3043–3050.

32 G. G. Krivov, M. V. Shapovalov and R. L. Dunbrack, *Proteins: Struct., Funct., Bioinf.*, 2009, **77**, 778–795.

33 D. Xu and Y. Zhang, *Proteins: Struct., Funct., Bioinf.*, 2012, **80**, 1715–1735.

34 E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng and T. E. Ferrin, *J. Comput. Chem.*, 2004, **25**, 1605–1612.

35 D. Xu and Y. Zhang, *Biophys. J.*, 2011, **101**, 2525–2534.

36 E. Krieger, K. Joo, J. Lee, J. Lee, S. Raman, J. Thompson, M. Tyka, D. Baker and K. Karplus, *Proteins*, 2009, **77**(Suppl 9), 114–122.

37 K. Pearson, *London, Edinburgh Dublin Philos. Mag. J. Sci.*, 1901, **2**, 559–572.

38 A. Shkurti, R. Goni, P. Andrio, E. Breitmoser, I. Bethune, M. Orozco and C. A. Laughton, *SoftwareX*, 2016, **5**, 44–50.