

Prediction of worm movement trajectory from known orientations of cilia

Model description

To test the hypothesis that orientation of the cilia can ultimately determine the movement phenotype of a worm as a whole, we devise a simple model that can predict worm movement basing on experimentally observed spatial and angular distributions of cilia within worms.

We simulate a worm as a solid body, meaning that there will be no deformation of a worm body (i.e. no stretching or bending): the body is translated and rotated as one solid unscalable entity. A body has rectangular shape and we divide it, as in our experiments, into segments; the head and the tail have the same geometry (Fig. 1A). We can arbitrarily set the length (L), the width (W) and the mass (M) of the whole body; the number of segments along L and W , from which the mass and the size of a single segment is then calculated.

Each segment will comprise a number of nodes (red circles in Fig. 1A) that represent rootlets of cilia, and with each node cilia beating force is associated during simulations. When a worm is being created, in each segment the number of the nodes N is drawn randomly from a normal distribution, the mean and the standard deviation of which we control (N_m , N_σ in Fig. 1A). Spatial distribution of these N nodes within a segment is uniform: there is no spatial bias and a node can be placed anywhere inside the segment with the same probability.

The parameters described above (worm mass, length, width; segment mass, length, width, number; nodes numbers and spatial distribution) are fixed during generation of a worm and do not change during simulation runs. Ideally, one segment should represent one cell with its unique spatial configuration of nodes. In our simulations, we construct a worm so that its segments conform to our experimental worm segmentation: 5 segments along the length and 10 segments along the width. However, this is absolutely not necessary to reproduce our simulation results.

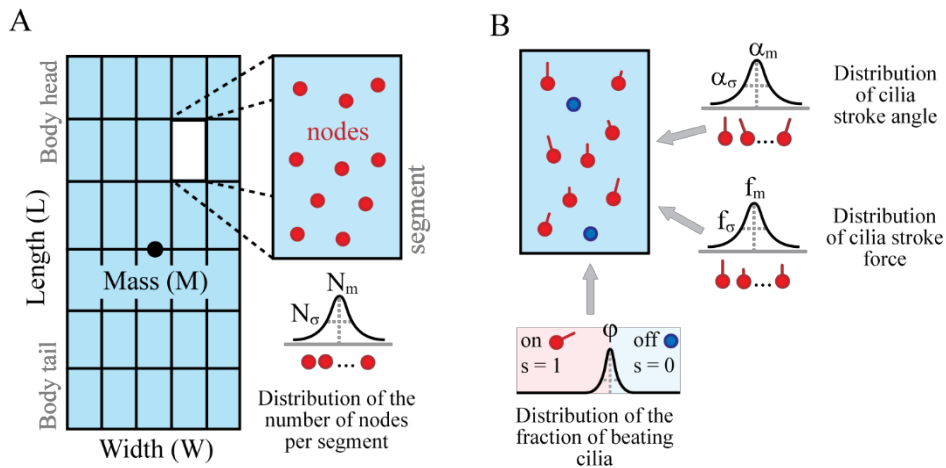


Figure 1. Structure of a simulated worm. (A): Worm body consists of several segments, each containing a number of nodes that are spatially uniformly distributed. The number of nodes in each segment is drawn from a normal distribution; (B): Configuration of beating cilia within each segment changes with new iteration; for this, parameters of cilia beating forces at each node are re-drawn from normal distributions (force angle, force magnitude, and on-off state).

During each simulation iteration nodes in all segments individually generate a force due to cilia beating; these forces vary in direction and magnitude. Therefore, we associate with each node the angle of the beating force α and its absolute magnitude f ; their values will be drawn from normal distributions, the mean and the standard deviation of which we control (Fig. 1B). The angle α is counted clock-wise from the Tail-Head axis for positive angles and counter clock wise for negative.

We also introduce notation of incoherency in cilia beating: at any given iteration, there is only a fraction of segment's nodes (ϕ_{on} , varies from 0 to 1) that have beating cilia and thus are "on" ($s = 1$, red nodes), whereas the other segment's nodes ($1 - \phi_{on}$) have resting cilia and are "off" ($s = 0$, blue nodes). Thus, when $\phi_{on} = 1$, cilia beating is fully coherent and all cilia beat together during an iteration. The fraction of nodes with beating cilia ϕ_{on} is drawn from a normal distribution, the mean and the standard deviation of which we control (Fig. 1B).

These three normal distributions (α , f , and ϕ_{on}) are constructed during generation of a worm, apply to all segments within this worm and do not change during simulation runs. During new iteration, each segment acquires a new configuration of beating cilia by drawing randomly parameters α , f , and s for each node from these distributions; each node is first set to "off" state and then turned "on" with the probability of ϕ_{on} .

Some parameters can be found experimentally: the worm sizes (L , W); normal distributions for drawing the number of nodes per segment (N) and the angular distribution of beating forces (α) (may be found experimentally

in each segment as in Figure X - reference to your data). Some we set arbitrarily: the worm mass (M), the magnitude distribution of the beating force (f) and the probability of on-off state (ϕ) are set arbitrarily.

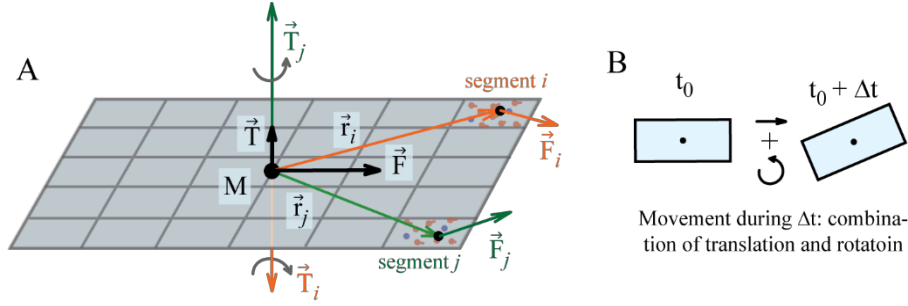


Figure 2. Worm movement. (A): Forces (\vec{F}_i, \vec{F}_j) and torques (\vec{T}_i, \vec{T}_j) arising due to cilia beating in segments i and j and their total effect on the worm center of mass (\vec{F}, \vec{T}); (B): Movement of the worm due to translation and rotation.

We define worm movement as a combination of translation of the whole body and rotation around its center of mass M (Fig. 2B). To find the translation, we first calculate the total force \vec{F} generated by all beating cilia in the worm (in all segments at each node); since the worm is modeled as a rigid body, all cilia beating forces may directly be applied to the center of mass M . We then model the experimentally observed constant speed movement of worms as follows. We assume that during each infinitesimal time span Δt the worm speed starts from 0, accelerates due to action of \vec{F} and decelerates back to 0 due to action of counter forces (e.g. friction). More specifically, in the first half (from 0 to $\Delta t/2$) acceleration is $a = F/M$ (according to second Newton's law), and during the second half (from $\Delta t/2$ to Δt) deceleration is $a = -F/M$ (we use symmetric evolution of the speed for simplicity of calculations). Such periodic movement can be approximated as movement with constant speed $V = \frac{F\Delta t}{4M}$. Thus, we can calculate the translation from old position p_0 to new position p during iteration time Δt as

$$\vec{p} = \vec{p}_0 + \frac{\vec{F}}{4M} \Delta t^2 \quad (1)$$

To find the rotation, we calculate for each i_{th} segment the net force \vec{F}_i arising from all its beating cilia and the torque $\vec{T}_i = \vec{r}_i \times \vec{F}_i$ about the worm's center of mass M that \vec{F}_i generates (Fig. 2A). The total torque of the worm \vec{T} is then calculated as the sum of all \vec{T}_i . According to Euler's second law: $\frac{\Delta \omega}{\Delta t} = \frac{\vec{T}}{I}$, where $\frac{\Delta \omega}{\Delta t}$ is the angular acceleration, $\omega = \frac{\theta}{\Delta t}$ is the angular speed, $I = \sum_i r_i^2 m_i$ is the moment of inertia of the worm body with r_i and m_i is the position and the mass of the i_{th} worm segment. Following the same considerations as in the case of translation, we can approximate worm's rotation as rotation with constant angular speed $\omega = \frac{T\Delta t}{4I}$. Thus, we can calculate the rotation from old angle θ_0 to new angle θ during iteration time Δt as

$$\theta = \theta_0 \pm \frac{T}{4I} \Delta t^2 \quad (2)$$

where the sign depends on the direction of the torque.

In each new iteration, all segments acquire new cilia beating configuration, after this new total force \vec{F} and total torque \vec{T} are calculated; then using equations (1) and (2) new position and new body orientation (\vec{p}, θ) are calculated.

Order of events in the simulation

- Worm generation. Parameters of the worm:
 - Body mass (M), length (L) and width (W), initial orientation (in Fig. 1 orientation is 90 degrees).
 - Segment numbers along body length and width N_L, N_W .
 - Node number bias; mean and standard deviation (N_m, N_σ).
 - Cilia beating angle bias; mean and standard deviation (α_m, α_σ).
 - Cilia beating force strength bias; mean and standard deviation (f_m, f_σ).
 - Cilia beating coherency bias; mean and standard deviation (ϕ_m, ϕ_σ).
 - Segment mass is calculated, node positions are set, cilia beating angles and forces are set, on/off state of each node is set.

- Simulation starts.
 - Iteration starts. Node positions never change.
 - Cilia beating angle changes randomly within the given bias.
 - Cilia beating force changes randomly within the given bias.
 - Cilia beating coherency is used to set randomly a fraction of nodes in off state.
 - In each segment the total cilia beating force and torque is calculated.
 - Total force and torque acting on the whole worm body is calculated
 - The whole body is moved according to the calculated force.
 - The whole body is rotated according to the calculated torque.
 - Repeat the iteration.

To test the simulation framework and see whether the results are qualitatively consistent, we run several simple tests. First, we generate two worms: one is biased to rotate left (Fig. 3A, worm 2, with orange segment) and the other is biased to rotate right (worm 1, with blue segment). All their segments have nodes generating force straight forward (grey segments), except for the colored two that have directional bias. Their trajectories are, as expected, mirrored, as shown with trajectory evolution and body orientation in Fig. 3A.

Second, we generate several worms with the same shape parameters, but one without noise and the rest with random noise; their trajectory evolution and body orientation (Fig. 3B) show the effect of noise. Varying and even opposite direction of rotation is caused by random asymmetrical distribution of nodes throughout the worm body (which is fixed after a worm has been generated), whereas the noise (visible in body orientation curves) comes from random cilia beating direction, force and on/off state.

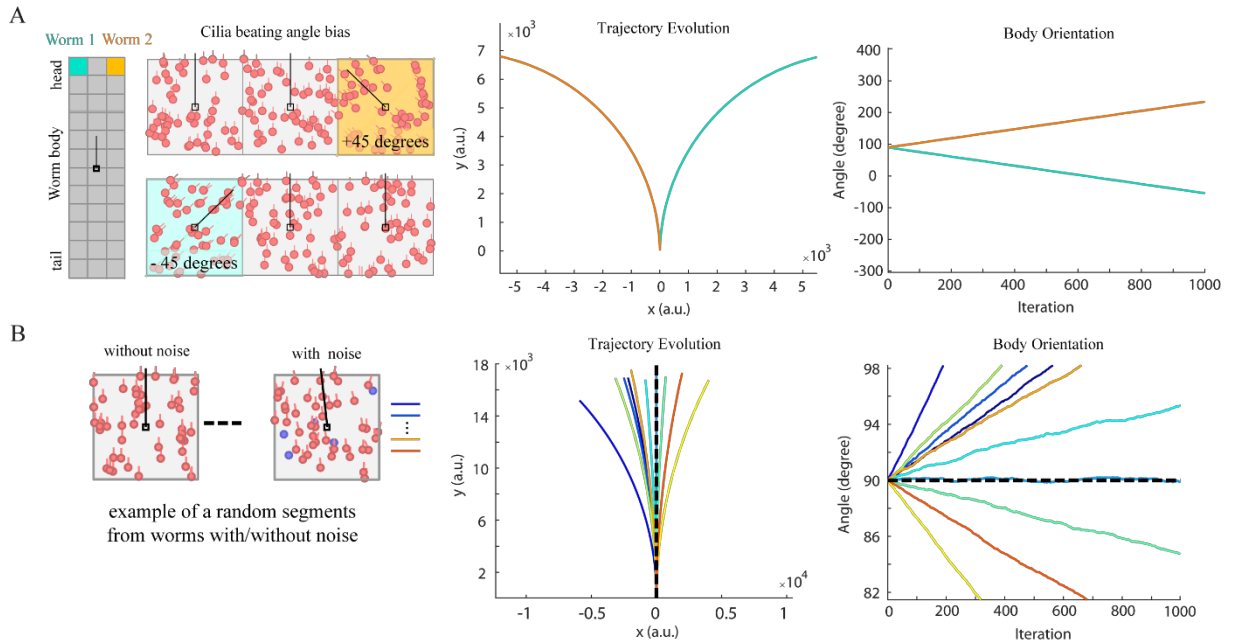


Figure 3. Test of the simulation framework. One worm without noise (standard deviations $\alpha_\sigma = f_\sigma = \varphi_\sigma = 0$) and 10 worms with noise (non-zero $\alpha_\sigma, f_\sigma, \varphi_\sigma$). (A): Initial orientation of any (noisy or not) worm is 90 degrees (counted counter-clock wise from x-axis), number of nodes per segment $N_m = 50$ with $N_\sigma = 5$, and cilia beating angle bias $\alpha_m = 0$ (along the body orientation). Example segments from two worms are shown: not noisy (above) and noisy (below); (B): Trajectory and (C): Evolution of body orientation during 1000 iterations. Not noisy worm is shown with dashed black line.

Next, we generate worms with exactly the same angular bias of cilia per segment as in the experimentally observed WT, mutants *odf2*, *vfl3*, and *vfl1* and run simulations on them. Our framework allows us qualitatively reproduce both linear translation on short time scale (Fig 4A-C, left column) and circular movement on longer time scale (Fig. 4 right column); the direction of translation and rotation are properly captured as well. Interestingly, even though these mutants translate in opposite directions, the both seem to circle (rotate) in the same direction.

However, with this simple model it is not possible to quantitatively reproduce the speed of rotation or the speed of translation; it allows only qualitative translation of experimentally observed angular distributions into trajectories. Consequently, it is not possible to compare two simulations with quantitatively (speed of run, speed of rotation etc.).

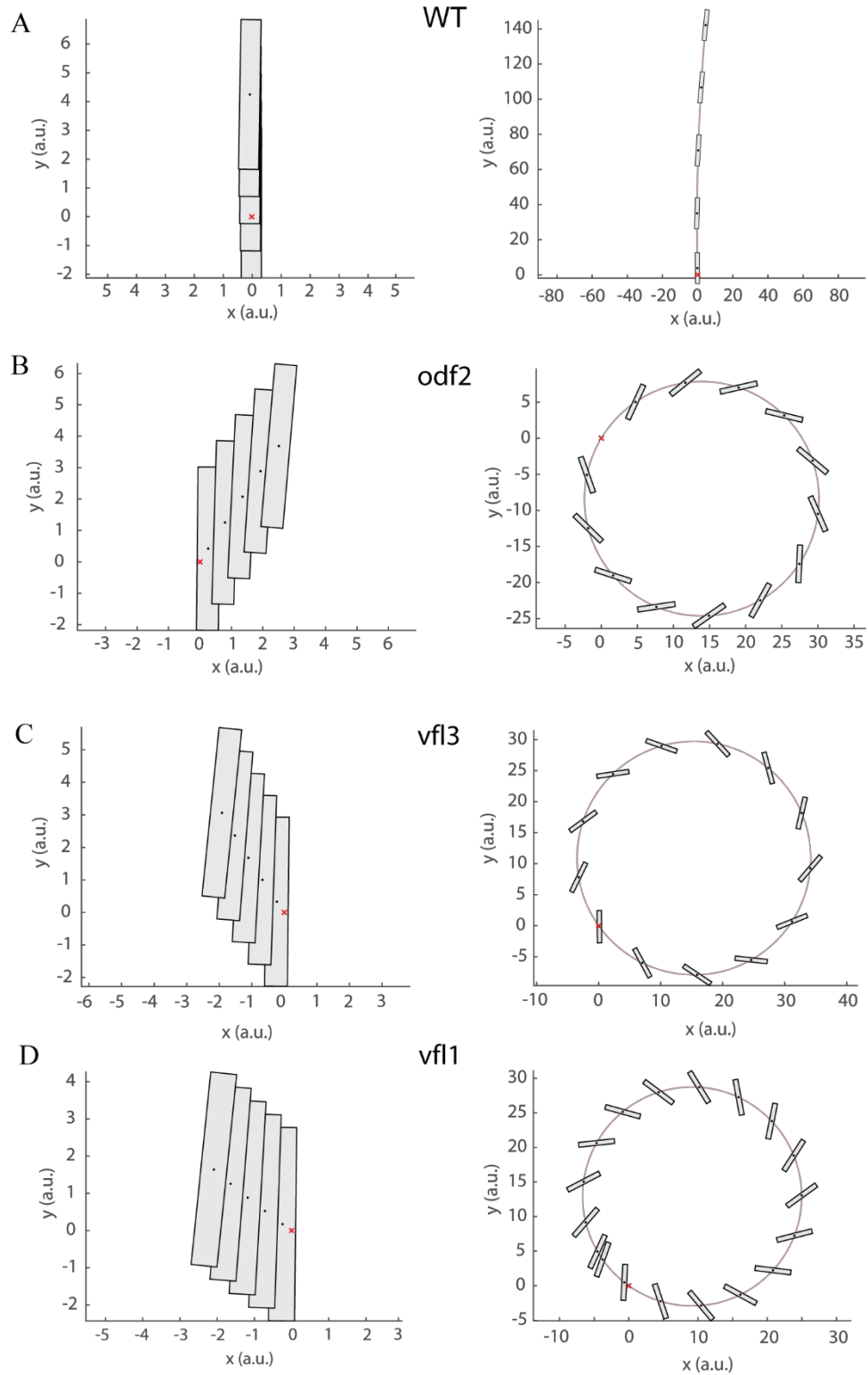


Figure 5. Simulated trajectories of WT and mutants odf2 and vfl3; cilia angular bias in each segment of the simulated worms was taken from experimental data. Left column: Trajectories on short time scales, right column: on long (~ 20 fold) timescales. Red cross at (0,0) indicates the starting point.