Patterns of molecular motors that guide and sort filaments

Beat Rupp and François Nédélec
EMBL, Meyerhofstr. 1, 69117 Heidelberg
nedelec@embl.de

September 7, 2012

Supplementary Material

Simulations

The physical scale of the system implies that Brownian motion plays a significant role whereas inertia can be neglected. Accordingly, the motion of filaments in Cytosim is calculated in discrete time steps with a Brownian dynamic approach, as described previously[15]. During a time-step, motors may also bind, move and unbind stochastically. The parameters used in this study are listed in Table 1 (see article). To make the simulation realistic, the filaments bear the measured characteristics of microtubules (MTs), and the properties of the motors are those of conventional kinesin. We believe that the main results should hold for actin filaments and their associated motors, if the patterns can be scaled down to compensate for the increased flexibility and lower duty ratio of the motors, but we have not run any simulation to evaluate this claim.

Microtubules

Microtubules are modeled as infinitely thin flexible oriented segments, thus neglecting protofilament structure. Bending elasticity obeys linear elasticity theory, and collisions between MTs are ignored[13]. The length of MTs is constant (5 or 10μm). MT mobility follows the theoretical mobility for an elongated cylinder moving near a planar surface[11]. Specifically, the drag coefficient was calculated as

$$\xi = \frac{4\pi\eta L}{\cosh^{-1}(1+h/r)}, \tag{1}$$

where η is the viscosity, L and r are the cylinder length and radius, and h is the height at which the closest cylinder surface travels above the surface. The drag force is then proportional to the speed of motion: $f = \xi\, v$. Values for h in gliding assays vary considerably, depending on how the motors are attached to the slide, and also on the method used to passivate the surface[16]. We used an experimental value $h = 15\,\mathrm{nm}$[13]. At this distance, the presence of the close surface effectively reduces the mobility of the filament by a factor ~ 6. Depending on the method used to pattern the motor proteins (once these are known), h can be adjusted, since the influence on the mobility becomes important if $h \ll r$.

Motors

The plus-end directed motor has the characteristic of conventional kinesin, and the minus-end directed motor is an imaginary kinesin moving towards the minus-end. This is done here for simplicity, since except of their direction of movement, the properties of the plus-end and minus-end directed motors are then identical. In the simulation, motors are represented by their anchorage location (a vector) and when they are bound to a MT by an abscissa along the MT. Unbound motor may bind stochastically at the specified binding rate, to any MT located within their binding range. If they bind, they do so on the closest location on the MT, which is the projection of the motor anchorage location. Motor-binding sites are not saturating, and several motors can thus bind at the same site on a MT. Unbinding is force dependent, and the force-velocity relationship is linear (see Table 1).

Conventional gliding assays

To verify our model parameters such as motor attachment, binding range and detachment rates, we compared it to experiments done by Fallesen *et al.* 2011, where the number of attached motors were estimated in a uniform gliding assay. The absolute number of molecules bound to the surface depends on the chemistry used for attachment. For the conventional assays, this was estimated to be in a range from less than 100 to more than 1000 motors/μm^2 [4,9–12]. However, molecules may bind to the surface in a non-favorable orientation and are then inactive. Thus the fraction of functional motors in gliding assays is now thought to be more than an order of magnitude smaller than the nominal number of bound protein [11]. The simulation correctly reproduces the experimentally measured distance between motor proteins that are bound to the gliding MT (Figure S1), when we used the estimated number of active motors (rather than the number of attached molecules) in the simulations. Unfortunately, we do not know the density that could be achieved when printing a line of motors, because that should depend on the method of microfabrication, and we set the linear density somewhat arbitrarily to 10 motors/μm.

Microtubule gliding on bands of motors

A second validation of the simulation and parameters comes from experiments [2], where the guiding of MTs transported on a band of densely covered kinesin were quantified (Figure S2). This test involves MTs moving on patterns of molecular motors, and allowed us to validate the simulation specifically for the goal set in this study.

In the simulations, MTs of length 5 μm, which corresponds to the average MT length in the experiment, were placed at various angles with their minus-end at the edge inside an area of kinesin motors with a density of 12 and 25 motors/μm^2 [5]. One second was then given to the motors to bind and equilibrate while their velocities was set to zero. The simulation then proceeded normally and guiding is considered successful if the MT travelled at least 5 μm along the edge with always at least one motor attached (Figure S2). Approach angles were varied randomly, and binned at 10° per interval. The probability of guiding for each interval was calculated as the ratio of successful guiding events to the total number of MTs (Figure S2B). Standard errors for the probabilities are $\mathrm{SE} = \sqrt{p(1-p)/m}$, where p is the probability of guiding and m is the total number of MTs considered in the interval.

Representation of patterns

A prerequisite to using a genetic algorithm was to define a meaningful yet simple representation of the pattern in terms of ‘genetic’ material. The complexity should be neither too high (eg. encoding the position of each individual motor) nor too low (only few patterns can be described). Yet there are many possibilities between these two limits, and we could have used an array of circular regions patterned with either motor type [17]. We however opted for lines of motors, because these were known to be able to guide filaments, as demonstrated experimentally [1,4].

A pattern is composed of segments of motor proteins immobilized on a 2D surface. The number of segments is fixed. Each segment is composed of only one type of motor. In addition to length and motor type, segments are parametrized with a radius of curvature, two coordinates for position and a rotation angle (Figure 1).

For some evolution trials (for example the one that produced pattern 2E), curvature was forbidden. The segments are placed anywhere within the surface area, but periodic boundary conditions ensure that cropping does not occur since the borders are wrapped in. The density of motors along a segment is constant (10 motors/μm). This corresponds roughly to the high densities achieved in conventional assay, but we do not know the density that could be achieved on lines, as this depends on the patterning method.

Parameter	Symbol	Allowed range
Position	X, Y	anywhere inside the simulated rectangle (Table 1)
Orientation	θ	$[0, 2\pi]$
Length	L	$[1, 12]\,\mu$m
Arc angle	Ω	$[0, 2\pi]$, the corresponding curvature is $C = \Omega/L$
Motor type	m	orange or blue

Genetic algorithm

Genetic algorithms (GA's), a class of global search heuristics, are used to solve optimization problems in different domains[7]. Genetic algorithm are inspired from the principles behind natural evolution, namely mutation, recombination (crossover) and selection, and reuse the same terminology in a very different context[6,8,14]. GA's have some distinct advantages over traditional search methods. For example unlike direct search methods such as gradient ascent, they have the ability to escape from local optima as the spatial footprint of the search is extended. This is important because real world search spaces are often multimodal, noisy and fraught with discontinuities. Another important advantage of GA's is that they do not depend on any *a priori* knowledge of the problem they are deployed to solve. Random changes to the candidate solutions will be evaluated with a fitness function to determine whether those changes produced an improvement. The main drawbacks of GA's are that it can be difficult to find a good genetic representation of all the possibilities that one wishes to explore, and it may also be hard to define a fitness function to represent the desired goal. We solved these problems as explained below.

Working principle of a genetic algorithm

Potential solutions are encoded as lists of numbers or 'chromosomes'. The algorithm is initialized by creating a pool of individuals, which all have randomized parameters (for each parameter the allowed range is specified). The evolutionary cycle is started by introducing random mutations to individuals' parameters (also called 'genes') and by performing crossover between chromosomes of two randomly chosen individuals. Then, each individual of such a population is evaluated by calculating its fitness value. Based on their fitness values, a new generation with the same number of individuals is created according to a selection procedure. In this way, the population size remains constant. Cycles composed of mutation, crossovers, selection are repeated, and the algorithm terminates in our case after a specified number of generations (Figure 1).

The genetic algorithm was empirically parametrized. We used a population of 100, a value that provides enough variability in our system and still allows the calculation of a generation in a reasonable amount of time. The values of the mutation and crossover rates are also results of trials and errors.

Mutation and Crossover

Mutations prevent the development of a uniform population incapable of further evolution. Each individual may mutate with a rate of 0.01 per generation. We found that mutation probabilities higher than 0.05 were negatively affecting the convergence of the GA. When mutation occurs, a new value for one of the parameter (eg. the length of the segment or its curvature) is chosen randomly within the allowed range for this parameter (see above).

When a potential solution to an adaptive problem is contained in the genetic pool of a given population, it may become active only after recombinations between individuals. A crossover operator should keep good partial solutions to increase overall fitness. As in earlier successful applications of genetic algorithms, a crossover rate of 0.7 was used[8,14]. This means that at every generation, as many random pairs as individuals present in the population are selected, and crossover is applied with a probability of 0.7 to each of these pairs. When crossover occurs, a position is randomly selected along the 'genome' and the two posterior segments are swapped.

Selection

We used a technique known as elitism selection. With elitism, a percentage of the fittest individuals of the current generation are directly placed into the new parent generation. We used 5 to 10 % of elitist selection, ensuring that 5-10 of the best chromosomes survive at each generation[3].

The other individuals necessary to complete the new generation are selected via tournaments. In tournament selection, the fitness of two randomly chosen individuals of the population is compared and with a high probability the individual with higher fitness is selected for the next generation. The pool of individuals stays constant during tournament selection, because selected individuals are put back and can be chosen again. It is important that this selection probability is not 1 (we used 0.7) in order to prevent premature convergence, an undesirable state where the GA is stuck in a local optimum.

Evaluation of fitness

A configuration for cytosim is generated by inserting the values encoded in a genome at specific positions in a template file. This configuration file is transmitted to a computation node of the cluster, where Cytosim reads and executes it. The disposition of the segment is read first, and motors of the specified type are randomly and uniformly distributed over each segment. MTs of fixed length are then added at random position and with random orientation. The simulation then proceeds, calculating the motion of the filaments subjected to Brownian agitation and forces produced by bound motors. Finally, after $T = 200s$, the simulation stops and the positions of the MTs are saved, from which the fitness f is derived. For directional transport (patterns on figure 2), one MT was used in each simulation, and the fitness simply reads:

$$f = -x/v_m T, \tag{2}$$

where x is the x-position of a filament after time T and v_m is the unloaded speed of the motor. x is the position of the center of the filament. Although the simulation runs with periodic boundary conditions, the x considered above is not periodic and reflects the total distance travelled. Thus, the fitness linearly increases with the distance travelled West. To search for robust directionality (figure 2G) or sorting patterns (patterns on figure 3), two MTs of length 5 or 10 μm were present in each simulations, and the fitness is proportional to the sum of two coordinates (eg. $f = (x_{10} - x_5)/2v_m T$). In all cases the fitness is normalized to fall within [-1, 1].

Stochasticity

The stochastic nature of motor binding, unbinding and Brownian motion in the simulations implies that the fitness assignment is also stochastic. In particular, the outcome of a simulation can depend strongly on the initial MT position and orientation, which are both random. This problem is largely solved by averaging 3 simulations with the same configuration, at the cost of linearly increasing CPU time. This is not equivalent to using more than one MT per simulation, because motors are limited, and saturation of binding could occur. Thus at each generation, 3 simulations were performed with only one MT (two MTs for length-sorting) and averaged. Further averaging occurs between successive generations in the course of evolution.

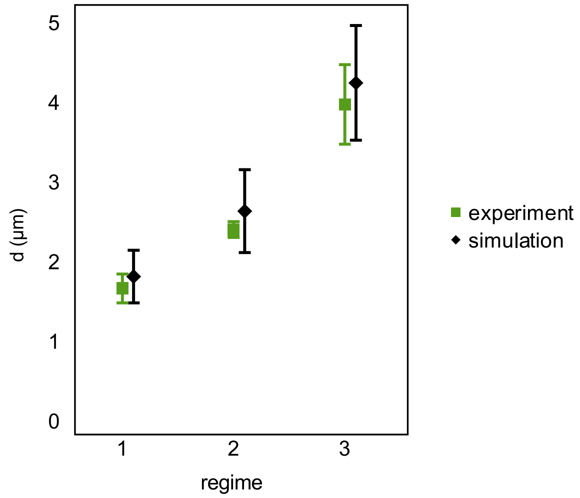
Implementation

The GA is implemented as a Python module that iteratively calls the Cytosim executable with the evolved sets of parameters. In order to optimize the algorithm's run time, it uses Apple's XGrid cluster technology to run on set of Macintosh computers. XGrid is accessed via the PyXG library (http://github.com/pyxg/pyxg, retrieved Nov 2, 2010) that provides a clean API for job submission, monitoring and manipulation from within Python.

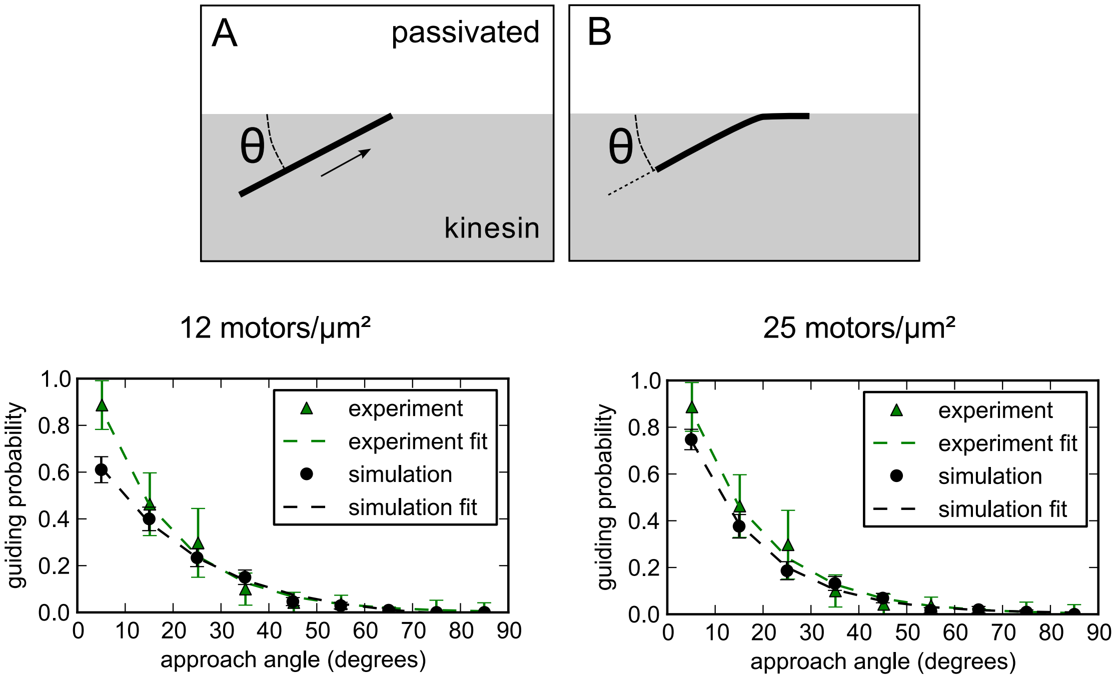
A single gliding assay simulation typically requires less than 10 seconds to compute in cytosim, but multiple runs were combined in one batch job, reducing the amount of network traffic and keeping the cluster nodes busy. When Xgrid returns the results from a generation, the fitness values are collected, individuals are combined and selected and the next round is then submitted.

Computation

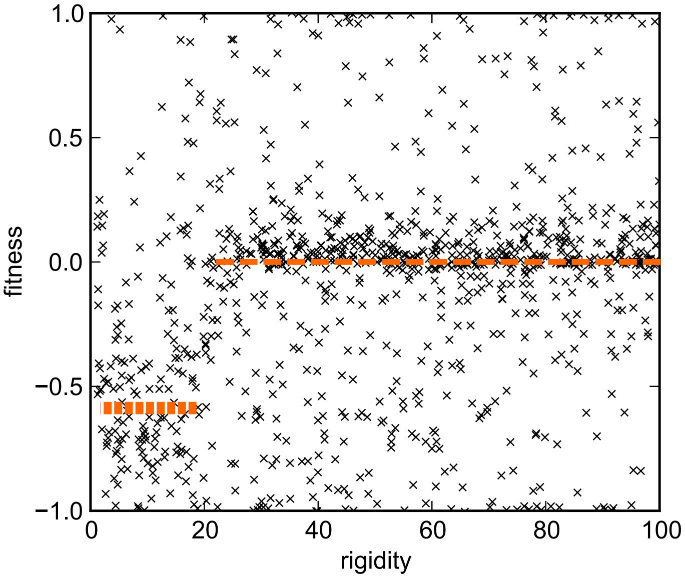
Each simulation required ∼15 seconds of CPU time using one core (sequential execution). Thus a complete evolution sequence as shown on Figure 1E, with 400 generations and 100 individuals, required 40,000 runs. Using Xgrid to address 2-5 desktop computers with 8 cores each, this took typically 1 day until completion. Our machines were Apple MacPro4,1 2x Quad-Core @ 3 GHz made in 2010 or similar.



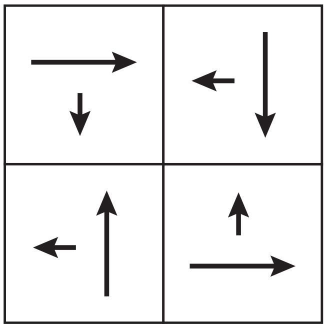
Supplementary Figure 1 Density of active motors in conventional assays. The simulation recapitulates the experimentally found distance *d* between motors bound to a MT in a gliding assay. Experimental data (in green) from Fallesen *et al.* 2011. In the simulation, a MT of length $10\,\mu$m was placed on a lawn of motors with a density σ of 25, 12 and 4.5 motors/μm^2 (regimes 1, 2 and 3) and was allowed to equilibrate for 10 seconds. The number of bound motors was averaged from 10 seconds of simulation. For each regime 50 simulations were averaged and plotted with standard deviation.



Supplementary Figure 2 Guiding microtubules with a band of motors. Top: A microtubule gliding on a uniform loan of motors reaches a passivated area at an angle Θ. If Θ is small enough, because the emerging MT tip wiggles due to thermal fluctuations, the MT can be guided along the boundary (B). Otherwise, the MT will proceed straight into the passivated area and eventually detach from the surface. **Bottom:** Probability of MT guiding for various approach angles, with experimental results from[2] and simulations (N=1000). MTs of length 5 μm were used. The dashed lines indicate the fit to experiments ($1.2\,e^{-0.08\,\theta}$) and simulations ($0.82\,e^{-0.05\,\theta}$ and $1.03\,e^{-0.07\,\theta}$ for motor densities of 12 and 25 motors/μm^2, respectively).



Supplementary Figure 3 The transport efficiency of pattern 3A for a filament of length 10 μm as a function of its rigidity (N=1000). Transport stops at $\kappa \sim 20$, when the filament becomes too stiff to fold.



Supplementary Figure 4 Four orthogonal sorting areas (Pattern 3E) can be combined to direct short filaments West, and long filaments East. Long (resp. short) arrows indicate the direction of motion of long (resp. short) filaments in each area.

References

[1] Maniraj Bhagawati, Surajit Ghosh, Annett Reichel, Klaus Froehner, Thomas Surrey, and Jacob Piehler. Organization of motor proteins into functional micropatterns fabricated by a photoinduced fenton reaction. *Angew Chem Int Ed Engl*, 48(48):9188–91, 2009.

[2] J. Clemmens, H. Hess, R. Lipscomb, Y. Hanein, K.F. Bohringer, C.M. Matzke, G.D. Bachand, B.C. Bunker, and V. Vogel. Mechanisms of microtubule guiding on microfabricated kinesin-coated surfaces: Chemical and topographic surface patterns. *Langmuir*, 19(26):10967–10974, 2003.

[3] Kenneth A De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, 1975.

[4] J.R. Dennis, J. Howard, and V. Vogel. Molecular shuttles: directed motion of microtubules along nanoscale kinesin tracks. *Nanotechnology*, 10:232–236, 1999.

[5] Todd L Fallesen, Jed C Macosko, and G Holzwarth. Measuring the number and spacing of molecular motors propelling a gliding microtubule. *Phys Rev E Stat Nonlin Soft Matter Phys*, 83(1 Pt 1):011918, Jan 2011.

[6] D B Fogel and L C Stayton. On the effectiveness of crossover in simulated evolutionary optimization. *Biosystems*, 32(3):171–82, 1994.

[7] D.B. Fogel. An introduction to simulated evolutionary optimization. *Neural Networks, IEEE Transactions on*, 5(1):3–14, 1994.

[8] David E Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.

[9] W.O. Hancock and J. Howard. Processivity of the motor protein kinesin requires two heads. *The Journal of cell biology*, 140(6):1395, 1998.

[10] H. Hess, J. Clemmens, C. Brunner, R. Doot, S. Luna, K.H. Ernst, and V. Vogel. Molecular self-assembly of “nanowires” and “nanospools” using active transport. *Nano Lett*, 5(4):629–633, 2005.

[11] A J Hunt, F Gittes, and J Howard. The force exerted by a single kinesin molecule against a viscous load. *Biophys J*, 67(2):766–81, Aug 1994.

[12] Parag Katira, Ashutosh Agarwal, Thorsten Fischer, Hsien-Yeh Chen, Xuwei Jiang, Joerg Lahann, and Henry Hess. Quantifying the performance of protein-resisting surfaces at ultra-low protein coverages using kinesin motor proteins as probes. *Advanced Materials*, 19(20):3171–+, October 2007.

[13] Jacob Kerssemakers, Jonathon Howard, Henry Hess, and Stefan Diez. The distance that kinesin-1 holds its cargo from the microtubule surface measured by fluorescence interference contrast microscopy. *Proc Natl Acad Sci U S A*, 103(43):15812–7, Oct 2006.

[14] M Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1998.

[15] F. Nedelec and D. Foethke. Collective langevin dynamics of flexible cytoskeletal fibers. *New J. Phys.*, 9:499–510, 2007.

[16] Guillaume Romet-Lemonne, Martijn VanDuijn, and Marileen Dogterom. Three-dimensional control of protein patterning in microfabricated devices. *Nano Lett*, 5(12):2350–4, Dec 2005.

[17] Vivek Verma, William O Hancock, and Jeffrey M Catchmark. Nanoscale patterning of kinesin motor proteins and its role in guiding microtubule motility. *Biomed Microdevices*, 11(2):313–22, Apr 2009.