Proteins at interfaces: Conformational behavior and wear

Emmanuel Louis Pierre Dumont

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2014

ABSTRACT

Proteins at interfaces:
Conformational behavior and wear

Emmanuel Louis Pierre Dumont

Proteins at interfaces play a major role in biomaterials[1,2] and lab-on-a-chip devices.[3,4] Protein interactions with the surface change their conformations and therefore their ability to bind to their respective ligands. Another major area of interest surrounding biomaterials and lab-on-a-chip devices is the prediction and prevention of wear.[5,6] Wear is the progressive loss of material from an object caused by contact and relative movement of the contacting solid, liquid, or gas.[7-9] It is estimated that wear costs 1% of the gross domestic product (approximately $150 billion for the US).[7,10] With the emergence of drug-releasing implants and lab-on-the-chip devices, wear has also become a major concern in bio- and nano- technology.[11] In our laboratory, we use microtubules (filamentous proteins) gliding on kinesin motor proteins as transporters in biosensors.[12] This system, known as the motility assay, is ideal for studying how the conformation of kinesins impacts the gliding of microtubules and therefore the performance of the biosensor. The proposed studies seek to show that kinesins' geometry changes with their grafting density following De Gennes' scaling laws[13] for flexible polymers (Chapter2 , published in Langmuir as E.L.P. Dumont, H. Belmas, and H. Hess, *Observing the mushroom-to-brush transition for kinesin proteins,* 2013, 29 (49), 15142-15145) and that microtubules experience molecular wear due to their repeated interactions with kinesins (Chapter 3, under review for Nature Nanotechnology as E.L.P. Dumont and H. Hess, *Molecular wear of microtubules propelled by surface-adhered kinesins*). These two results permit the prediction of the lifetime of biosensors using kinesin-propelled microtubules (Chapter 4, to be submitted to Nano Letters as Y. Jeune-Smith, E.L.P. Dumont and H. Hess, *Wear and breakage combine to mechanically degrade kinesin-*

powered molecular shuttles). I also discuss the importance of mechanical fatigue for molecular machine design (Chapter 5, published as H. Hess and E.L.P. Dumont, *Fatigue Failure and Molecular Machine Design*, Small, 7, 1619-1623, 2011).

Finally, and it is unrelated to the previous chapters, I developed Monte Carlo simulations for protein adsorption on polymer-coated surfaces (Chapter 6, to be submitted as E.L.P Dumont, A.V. Guillaume, A. Gore, and H. Hess, *Random Sequential Adsorption of proteins on polymer-covered surfaces: A simulation-based approach*) and I explored a molecular model to explain the fracture of materials at low stresses (Chapter 7).

Chapter 1: ***Microtubules, kinesins, and their interactions.***

In Chapter 1, I introduce the microtubules and kinesins, found in all living cells, and how they interact. We also introduce how these proteins are used in the "motility assay", where microtubules glide on a kinesin-coated surface. Finally, we present existing litterature of molecular nanotribology (friction and wear) of microtubules and kinesins.

Chapter 2: ***Observing the mushroom-to-brush transition for kinesin proteins.***

The height of polymers grafted to a surface is predicted to be constant at low densities ("mushroom" regime) and increase with the third root of the polymer surface density at high densities ("brush" regime). This mushroom-to-brush transition is explored with kinesin-1 proteins adhered to a surface at controlled densities. The kinesin height is measured by attaching fluorescently labeled microtubules to the kinesins and determining their elevation using fluorescence interference contrast microscopy. Our measurements are consistent with a mushroom and a brush regime and a transition near the theoretically predicted density. The

mushroom-to-brush transition may play a role in protein behavior in crowded cellular environments and be exploited as a signal in intracellular regulation and mechanotransduction.

Chapter 3: ***Molecular wear of microtubules propelled by surface-adhered kinesins.***

Wear, the progressive loss of material from a body caused by contact and relative movement, is a major concern not only in engineering but also in biology.[8,10,14,15] Advances in nanotechnology both enable the study of the origins of wear processes at the atomic and molecular scale and demand the prediction and control of wear in nanoscale systems.[11,16,17] Here, we show that wear occurs in an in vitro system consisting of microtubules gliding across a surface coated with kinesin-1 motor proteins, and that energetic considerations suggest a molecule-by-molecule removal of tubulin proteins. The wear rates show a complex dependence on sliding velocity and kinesin density, which – in contrast to the friction behavior between microtubules and kinesin[18] – cannot be explained by simple chemical reaction kinetics.

Chapter 4: ***Wear and breakage combine to mechanically degrade kinesin-powered molecular shuttles.***

In this Chapter, I show how the combined wear of microtubules propelled by surface-adhered microtubules (Chapter 3) and their breaking (an already known phenomenon) permit the prediction of the failure of microtubule-based biosensors, similar to the failure of macroscopic machines. In macroscopic machines, failure as a result of activation is the result of breakage or wear. Breakage is a sudden and permanent phenomenon often caused by fatigue. Wear, the gradual removal of small amount of material, causes an increasing deviation of the part

dimensions from the ideal. Unless breakage intervenes, any system will ultimately fail due to wear. Reducing breakage and wear is a major consideration in machine design.[19]

Chapter 5: ***Fatigue Failure and Molecular Machine Design.***

Sophisticated molecular machines have evolved in nature, and the first synthetic molecular machines have been demonstrated. With our increasing understanding of individual operating cycles, the question of how operation can be sustained over many cycles comes to the forefront. In the design of macroscale machines, performance and lifetime are opposing goals. Similarly, the natural evolution of biological nanomachines, such as myosin motor proteins, is likely constrained by lifetime requirements. Rather than bond rupture at high forces, bond fatigue under repeated small stresses may limit the mechanical performance of molecular machines. Here we discuss the effect of cyclic stresses using single and double bonds as simple examples and demonstrate that an increase in lifetime requires a reduction in mechanical load and that molecular engineering design features such as polyvalent bonds capable of rebinding can extend the bond lifetime dramatically. A universal scaling law for the force output of motors is extrapolated to the molecular scale to estimate the design space for molecular machines.

Chapter 6: ***Random Sequential Adsorption of proteins on polymer-covered surfaces: A simulation-based approach***

Non-fouling polymeric coatings enable the suppression of protein adsorption to surfaces, and their perfection is the objective of many recent experimental studies.[20-25] Obtaining a theoretical understanding of the functioning of these coatings and the prediction of residual protein adsorption as a function of the coating properties has similarly attracted significant interest.[26-35]

A recent study developed a basic Random Sequential Adsorption (RSA) model for protein adsorption to non-fouling coatings, which was analytically solvable and yielded encouraging agreement with published experimental data.[36] The model assumed that polymer chains on the surface can be represented by hard spheres with a radius equal to their radius of gyration. These randomly distributed hard spheres obstruct the adsorption of proteins, again represented as hard spheres with a diameter equal to the diameter of the protein. The evolution of the protein density on the surface was calculated from the independent probabilities to penetrate the layer of adsorbed proteins and the layer of polymer chains. Here, we scrutinize this analytical model by conducting computer simulations of the adsorption process under the same assumptions. We find that the results of the computer simulations deviate significantly from the analytical solution, which indicates that the spatial distributions of proteins and polymers cannot be considered independently.

Chapter 7: ***A molecular model to explain material fracture at low stresses.***

In this Chapter, I extend Zhurkov's work on predicting the lifetime of materials with a simple molecular model.[37] In 1965, Zhurkov introduced a model[37] to predict the lifetime of materials under uniaxial tension. Zhurkov's model connects the lifetime of the material and the uniaxial stress (macroscopic experimental values) to several microscopic constants such as the enthalpy of sublimation and the Boltzmann constant. I extend this model by introducing the possibility of rebinding. This new model enables the prediction of the lifetime of material at low stresses.

Table of Contents

List of figures

Acknowledgments

I would like to thank Professor Henry Hess for supervising my research over my 4 years in his laboratory. I am also very grateful to Professor Nicolas Biais, Professor Christopher Jacobs, Professor Lance Kam, and Professor Samuel Sia for being on my dissertation committee. I would like to also thank former and current members of Henry Hess' laboratory: Ashutosh Agarwal, Rodney Agayan, Megan Amstrong, Corina Curschellas, Arnaud Guillaume, Siheng He, Ofer Idan, Parag Katira, Amy Lam, Amit Sitt, and Yoli Jeune-Smith. Finally, financial aid from the National Science Foundation under grant CMMI-0926780 and from the Liu-Ping fellowship granted by the department of Biomedical Engineering at Columbia for one year is gratefully acknowledged as well.

Forsan et haec olim meminisse iuvabit

(Perhaps someday it will help)

Virgil, Aeneid, Book 1

CHAPTER 1

Microtubules, kinesins, and their interactions

Be patient, for the world is broad and wide

Friar Laurence in Romeo and Juliet (Shakespeare)

Microtubules are filamentous proteins and are part of the cell cytoskeleton.

Eukaryotic cells have an ensemble of protein filaments – microfilaments, intermediate filaments, and microtubules – called the cytoskeleton. The cytoskeleton is used by cells to interact mechanically with their environment, for intra-cellular transport, and for mitosis.[38] Microtubules, which we use in our laboratory, are long and hollow cylinders made of the dimeric tubulin proteins (**Figure 1**). Their outer diameter is 25 nm and their inner diameter is 12 nm.[38] Like the other cytoskeletal filaments, microtubules are very dynamic, undergoing phases of growing and shrinking.

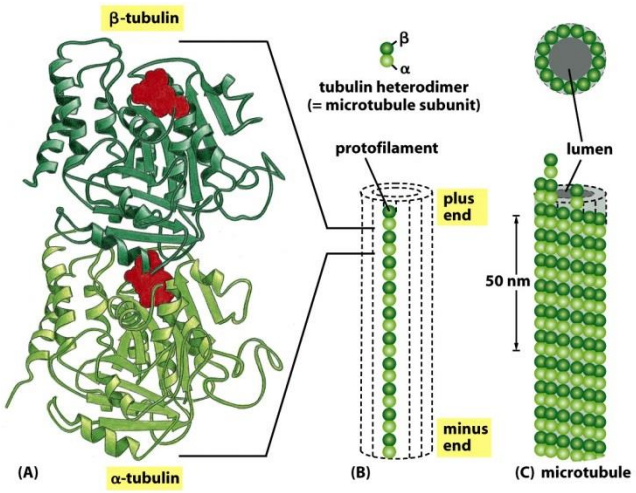

Figure 1. The structure of a microtubule and its subunit. (A) The subunit of each protofilament is a tubulin heterodimer, formed from a very tightly linked pair of α- and β-tubulin monomers. (B) One tubulin subunit (αβ heterodimer) and one protofilament are shown schematically. Each protofilament consists of many adjacent subunits with the same orientation. (C) The microtubule is a stiff hollow tube formed from 8-19 protofilaments aligned. From Alberts et al.[38]

Microtubules consist of αβ heterodimers of tubulin. These heterodimers are very stable and can be dissociated only by harsh treatments. Microtubules are polar structures: the fast-growing end is called the plus end and the slow-growing end is called the minus end. Most microtubules have 13 protofilaments, but microtubules can be formed with 8 to 19 protofilaments.[39]

Kinesins are motor proteins that walk along microtubules.

Motor proteins derive their name from their ability to generate forces, and are responsible for muscle contraction and the crawling and swimming of cells. They are also capable of intra-cellular transport of vesicles and of chromosomes during mitosis.[38] There are 3 major families of motor proteins: myosins, dyneins, and kinesins. Within each family, proteins share approximately 50% of their amino acid sequence with each other. There are more than two dozen types of kinesin proteins. The first kinesin motor was discovered in 1985 by M. Sheetz,[40] a professor of biology at Columbia University. Kinesins have the ability to bind to αβ heterodimers and, as a result, are able to walk on microtubules by hydrolyzing ATP molecules. Generally speaking, kinesins are processive motors, traveling for hundreds of steps along a microtubule before detaching.[38] Among the different families of kinesins, kinesin-1 proteins, which we use in our laboratory, are transport proteins: their main role was identified to be transporters of vesicles within cells. In contrast, kinesin-8 proteins play a role in microtubule depolymerization, which kinesin-1 proteins do not have.[41]

In August 2013, Gigant et al. revealed for the first time the details of the binding between kinesin-1 proteins and αβ heterodimers (**Figure 2**).[42] The technical hurdle was to obtain protein crystals without tubulin polymerization.

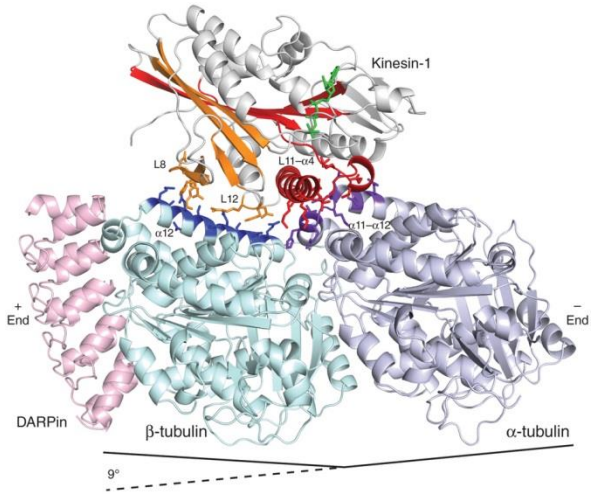

Figure 2. Structure of the kinesin-1 motor domain in complex with αβ-tubulin. Side chains involved in binding contacts (within 4 Å) are shown. The observed 9° curvature of tubulin subunits with respect to a linear microtubule is indicated. Adapted from Cochran et al.[43]

This high resolution of the binding mechanism between kinesin-1 and αβ heterodimer reveal a 9° curvature (**Figure 2**), as opposed to a straight conformation as found in the microtubule lattice, suggesting that the crystallization of the kinesin-heterodimer complex is not completely similar to what is found in cells. However, it is expected that this X-ray crystallography by Gigant and colleagues will lead to new insights about the mechanics of kinesin-1 proteins.

Fifteen years ago, it was discovered that kinesin proteins take 8-nm steps on microtubules for each ATP molecule they hydrolyze.[44] The way kinesins make their 8-nm steps, however, was a long standing controversy. Until 10 years ago, two models were still competing: the hand-over-hand mechanism and the inchworm movement (**Figure 3**).[45] In 2004, Yildiz and colleagues showed that kinesins walk hand over hand on microtubules.[46]

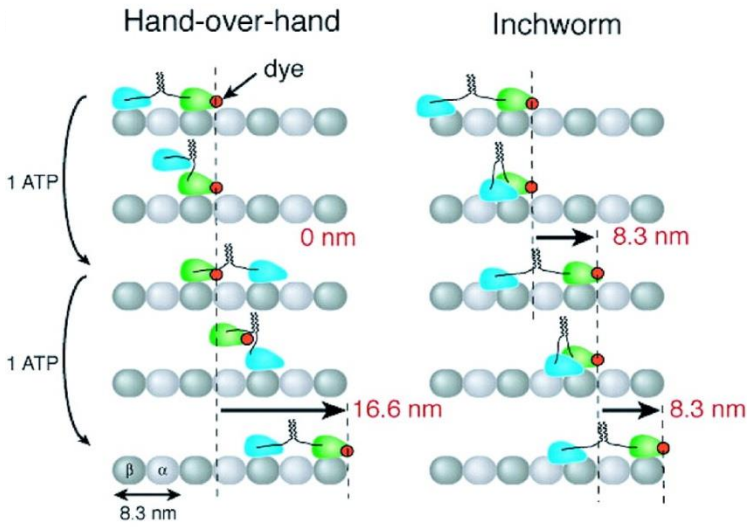

Figure 3. Examples of two alternative classes of mechanisms for processive movement by kinesin. The hand-over-hand model (left) predicts that a dye on the head of kinesin will move alternately 16.6 nm, 0 nm, 16.6 nm, whereas the inchworm mechanism (right) predicts uniform 8.3-nm steps. Adapted from Yildiz et al.[46]

Using kinesins and microtubules in a motility assay.

In 1985, for the first time, kinesins and microtubules were used by Vale et al. in the reverse geometry where kinesins are adsorbed to a surface and propel microtubules (**Figure 4**).[47] The kinesin conformational changes when adsorbing to the surface were not studied but Vale et al. noticed that kinesins kept their motor activity when adsorbed on a glass surface.

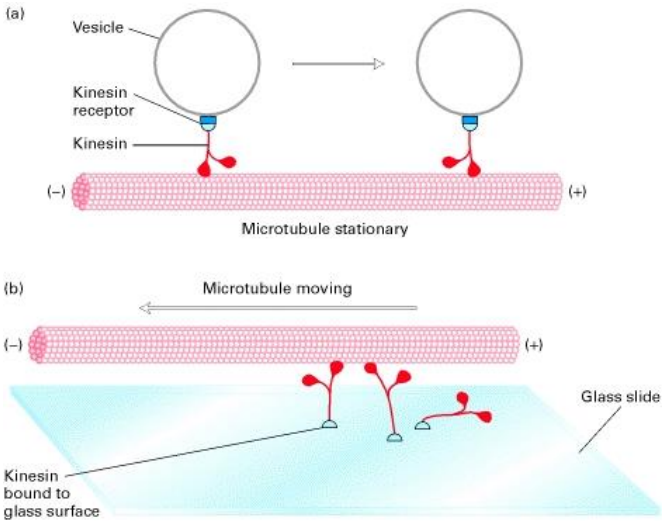

Figure 4. Sketch of a motility assay. (a) Kinesin-motored transport of vesicles along immobile microtubules. The kinesin molecules, attached to unidentified receptors on the vesicle surface, transport the vesicles from the (−) to the (+) end of a stationary microtubule. (b) Kinesin-catalyzed movement of microtubules. The kinesin molecules bound to the glass surface move toward the (+) end of the microtubule. Because the kinesin molecules are immobilized onto the coverslip, the sliding force is transmitted to the microtubule, which then moves in the direction of its (−) end. ATP is required for movement in both cases. Adapted from Lodish et al.[48]

In our laboratory, microtubules are first polymerized and then stabilized in paclitaxel (taxol) to avoid depolymerization.[49,50] When preparing the motility assay, the glass surface is first coated with casein proteins to avoid denaturing of kinesin-1 proteins when they land on the surface (**Figure 5**).[51]

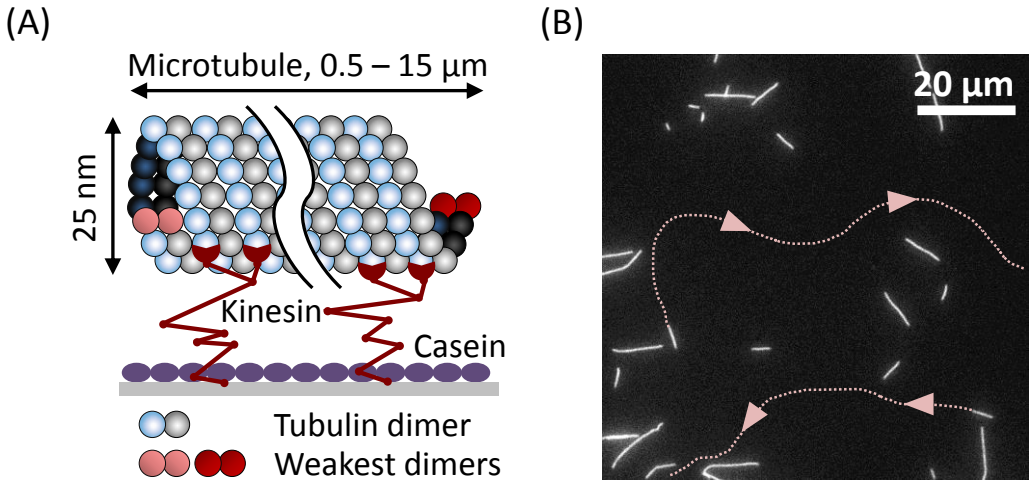

Figure 5. Sketch and picture of a motility assay. (A) Caseins are adsorbed to the surface before kinesins. Microtubules are usually 0.5 – 15 µm long, are stabilized with taxol (not shown in the drawing), and are fluorescently-labeled with rhodamine (not shown in the drawing). On both extremities, there are two weakly bound dimers that play an important role in the wear process described in the research plan and results. (B) Picture of a motility assay with 100x magnification. Microtubules glide all over the surface, as shown by the two traces.

Motility assays have been used to create biosensors in our laboratory.[12] As a result, the degradation of microtubules, which act as transporters in the biosensor, is critical to the lifetime of the biosensor. It was already known that microtubules break when they glide on kinesins.[52] However, the high number of binding events between kinesins and microtubules may induce also wear (our hypothesis for the 2nd specific aim). To measure wear, it is critical to measure the length of microtubules with the highest accuracy possible. However, using a magnification of 100x, a pixel represents a distance of 80 nm. As a result, any measurement "by hand" of the length of a microtubule is made with an uncertainty of at least a few pixels (**Figure 6**) since it is difficult to locate the extremities of the microtubule. Such an uncertainty is equivalent to

approximately half a micrometer and complicates the measurement of the wear process significantly. Fortunately, in 2011, Ruhnow and colleagues developed a MATLAB code to measure the length of filaments with sub-pixel accuracy (**Figure 6**).[53] We used Ruhnow's code to measure the wear of microtubules (See Research plan and results).

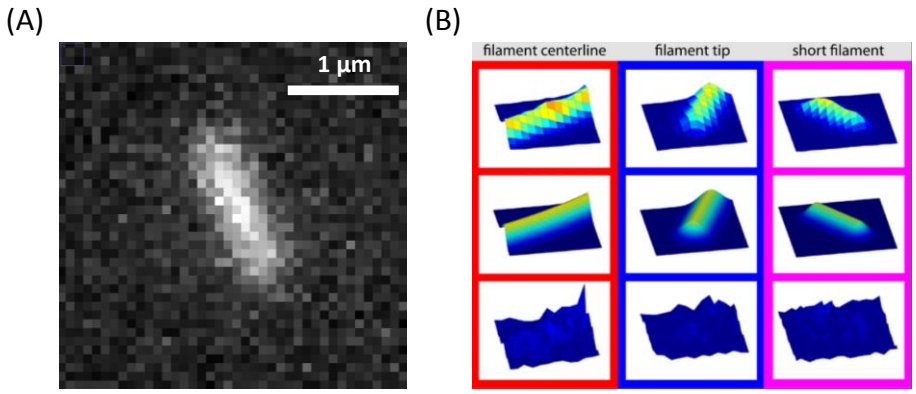

Figure 6. Measuring the length of microtubules. (A) Picture of a microtubule under 100x magnification. It is very difficult to distinguish the extremities of the microtubule. (B) Principles of the algorithm to determine the length of filaments with sub-pixel accuracy. (B) is adapted from Ruhnow et al.[53]

Molecular crowding of kinesins on microtubules.

Because they retain their motor activity, kinesins on a surface are the ideal testbed to study the impact of their crowding over their conformational behavior. Molecular crowding is an important phenomenon responsible for several age-related diseases such as Alzheimer's and the Parkinson's diseases.[54] In cells, molecular crowding of kinesins can create a traffic jam of kinesins on microtubules, as shown by Leduc et al. (**Figure 7**).[55]

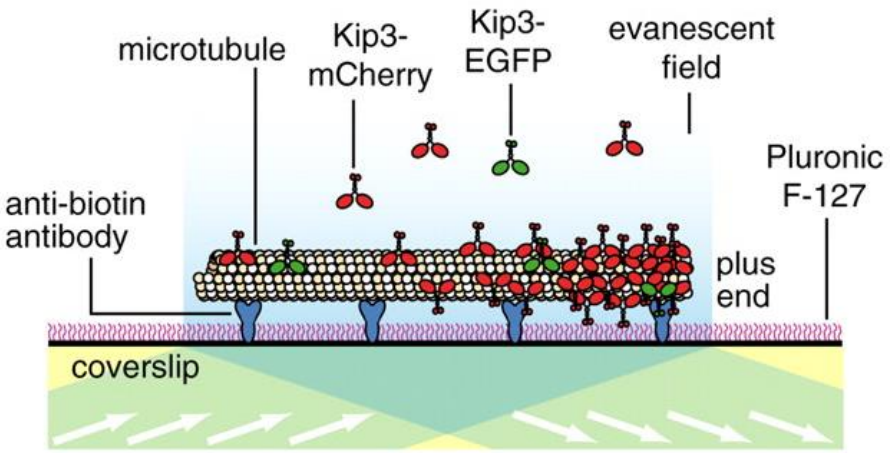

Figure 7. Molecular crowding on kinesins on a microtubule. Leduc et al. showed that molecular crowding of kinesins creates traffic jam. They never explored, however, the conformational change of kinesins when they are in a crowded environment. Adapted from Leduc et al.[55]

Scaling laws for flexible polymers.

More than 30 years ago, Alexander and De Gennes derived scaling laws which showed that the height of grafted flexible polymers is independent of grafting density at low densities ("mushroom" regime) and scales with the third root of the grafting density at high densities ("brush" regime).[13,56]

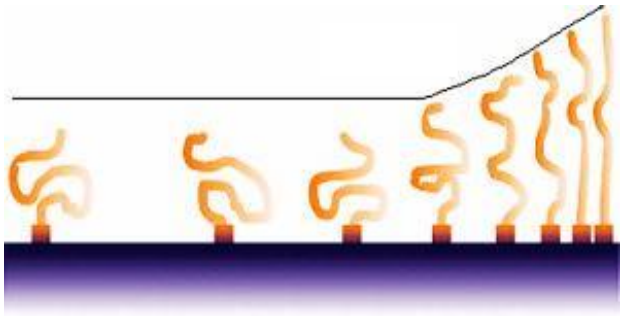

Figure 8. Sketch of the mushroom-to-brush transition for flexible polymers. On the left side of the drawing, polymers are in the mushroom regime and on the right side, they are in the brush regime. In the brush regime, the height of flexible polymers scales with the third root of the grafting density. Adapted from Brittain et al.[57]

The transition between the mushroom and the brush regime occurs when polymer chains start to overlap.[13,56,58] This scaling law was experimentally verified by Wu et al. who observed the mushroom-to-brush transition for grafted polymers using polyacrylamides (average molecular weight: 17 kDa; radius of gyration: 5.4 nm), as shown in the figure below reproduced from Wu's paper.[59]

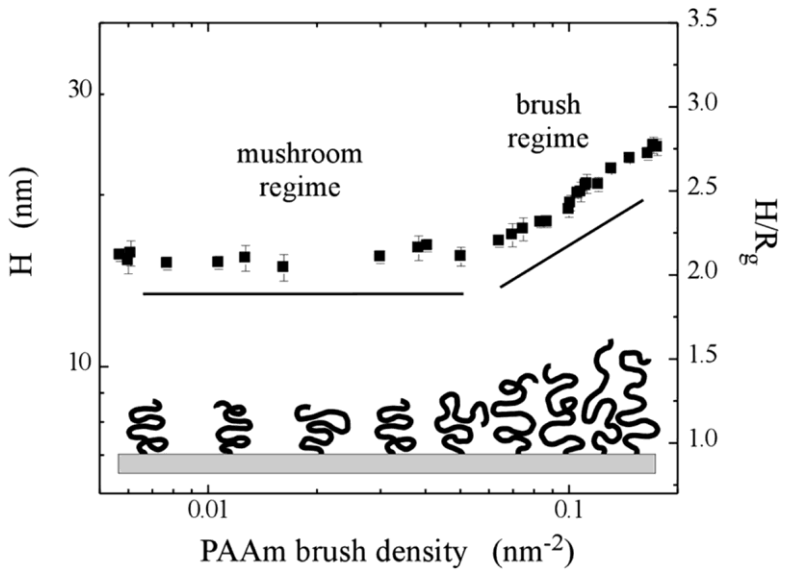

Figure 9. Mushroom-to-brush transition for a flexible polymer. Wu et al. were the first to observe the transition from the mushroom to the brush regime for flexible polymers. Adapted from Wu's paper.[59]

As predicted, the height of polymers in the mushroom regime (18 nm) was constant and the height increased with an exponent of roughly 1/3 (0.37 – 0.4) in the brush regime. The transition from the mushroom to the brush regime occurred when the available area per chain occurred at a reduced grafting density $\Sigma = 6$, where $\Sigma = \sigma\pi R_g^2$, σ is the kinesin grafting density, and R_g is the kinesin radius of gyration. This is significantly higher than the transition at $\Sigma = 1$ one would expect if polymer chains occupied half-ellipsoids of radius Rg and height 2Rg.[60] This "late" transition has been repeatedly observed, sometimes even at $\Sigma > 10$.[57]

Wear at the nanoscale.

Besides conformational changes, wear and friction are also of interest for lab-on-the-chip devices. These two phenomena were discovered and, as a result, defined for the macroscale.

They are part of the field of tribology, founded in 1966 by Jost,[61] which also encompasses the study of lubrication.[8]

Wear is the progressive loss of material from a body caused by contact and relative movement of a contacting solid, liquid, or gas.[7-9] Most of the time, wear results from the contact of asperities between two surfaces. Several empirical laws are used to describe wear. The most widely used equation is Archard's law of adhesive wear[62] where the material loss per time Ψ depends linearly on the relative speed v and the normal force F_L:

$$\Psi = k_w \frac{F_L v}{H}$$

where $k_w$is the wear coefficient and H is the hardness of the material. There have been attempts to adapt this law to the nano- and micro- scales, but these laws remain empirical and do not account for all wear phenomena at the nano- and micro- scales.[63] This year, Jacobs and Carpick reported an atom-per-atom wear of a silicon tip gliding on a diamond surface, showing that wear at the nanoscale is not necessarily a plastic phenomenon and does not result from fracture; rather it is the gradual removal of the most weakly bonded atoms.[17] In that paper, they moved away from empirical models and used the Bell equation to explain their experiments:

$$\Gamma = \Gamma_0 \exp\left(-\frac{\Delta G_{act}}{k_B T}\right)$$

Where Γ is the rate of atom loss due to wear, Γ_0 is a pre-factor (an effective attempt frequency), ΔG_{act} is the free energy of activation, k_B is Botzmann's constant, and T is the absolute temperature.

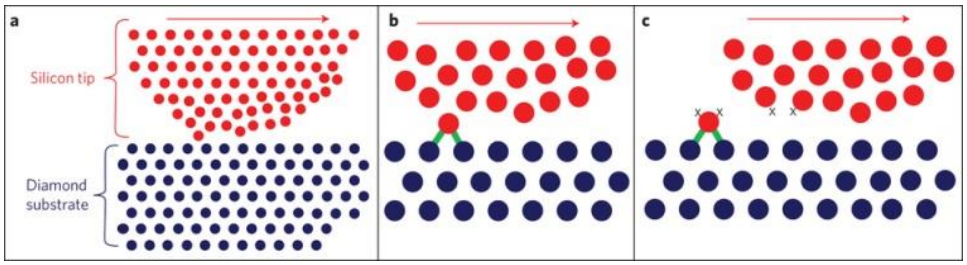

Figure 10. Proposed mechanism of nanoscale wear by stress-assisted atom-by-atom removal. (a) A sharp silicon asperity in adhesive contact with a diamond substrate slides in quasi-static equilibrium relative to the substrate. (b) One of the low-coordinated atoms from the asperity surface forms a covalent bond (green) to one or more substrate atoms. This thermally activated bond formation is facilitated by the compressive stress in the contact. (c) The bonds to the tip are then broken as the asperity slides on, leaving a silicon atom on the substrate. Adapted from Jacobs et al.[17]

The depolymerization of microtubules facilitated by kinesin-8, demonstrated by Varga et al.,[41] is conceptually different from the gradual wear of the silicon. In this case, the kinesin-8 proteins act collectively to depolymerize the plus end of the microtubules, targeting the most weakly bonded tubulin dimer. However, this behavior is considered targeted disassembly, which is conceptually different from wear, which is an undesired result of stresses caused by the normal operation of a mechanical system leading to system failure. Instead of kinesin-8, our laboratory uses kinesin-1, which has not been shown to exhibit targeted disassembly, thus making it suitable for studies on wear and friction.

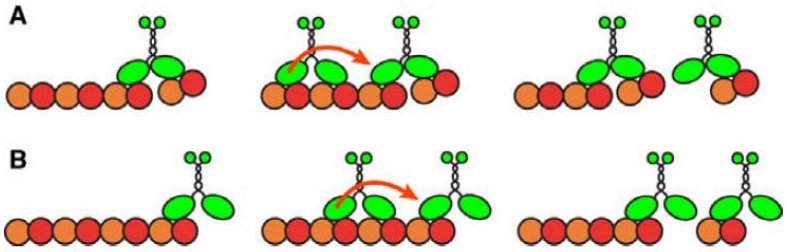

Figure 11. The two proposed mechanisms for the disassembly of microtubules by Kinesin-8 proteins. (A) Tubulin removal before bump-off: After a kinesin-8 protein reaches a vacant end, it severs the interdimer bond, perhaps due to intramolecular strain developed when both motor domains are tightly bound (left). An incoming kinesin-8 protein then bumps off the end-bound one. This acceleration of kinesin-8 detachment from the plus end occurs due to competition for the kinesin-8 binding site on the penultimate tubulin dimer (middle). The binding of the second kinesin-8 protein to the end is then followed by another severing (right). (B) Tubulin removal during bump-off: A kinesin-8 protein binds stably to the terminal tubulin dimer at the microtubule's plus end (left). An incoming kinesin-8 protein bumps into the paused molecule (middle) causing severing and dissociation of a kinesin-tubulin complex (right). Adapted from Varga et al.[41]

Friction at the nanoscale.

Friction is the resistance to motion during sliding or rolling that is experienced when one solid body moves tangentially over another with which it is in contact.[64] The resistive tangential force, which acts in a direction directly opposite to the direction of motion, is called the friction force. There are two main types of friction that are commonly encountered: dry friction and fluid friction. As its name suggest, dry friction, also called "Coulomb" friction, describes the tangential component of the contact force that exists when two dry surfaces move or tend to move relative to one another. Fluid friction describes the tangential component of the contact force that exists

between adjacent layers in a fluid that are moving at different velocities relative to each other as in a liquid or gas between bearing surfaces. Similar to wear, a few friction models, such as the Tolimson model, were made but they describe incompletely and sometimes unsuccessfully phenomena of friction at the nano- and micro- scales.[63]

In 2009, Bormuth and colleagues showed that when a kinesin-8 protein is dragged along a microtubule in absence of ATP, it exerts a dragging force, conceptually understood as "protein friction" (**Figure 12**).[18] They showed that the friction force is proportional to the sliding velocity at velocities up to 1 µm s^{-1}. They also showed that the energy barrier of this process is 13 ± 2 k_BT.

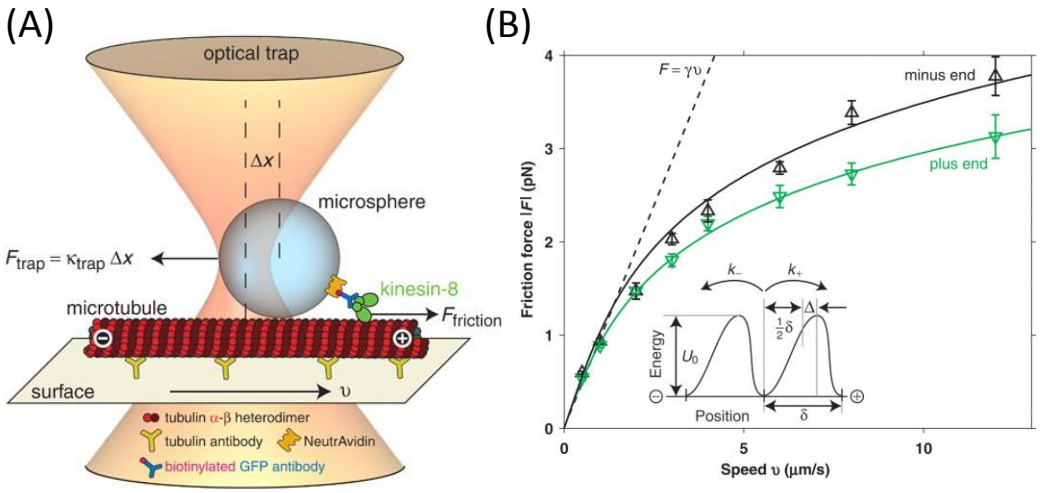

Figure 12. Schematic of the kinesin friction experiment. (A) Using a focused laser, a kinesin-8 coated microsphere is trapped close to an immobilized microtubule (not drawn to scale). Moving the stage with constant velocity past the stationary laser drags the microtubule lattice underneath the kinesin-8 molecule, creating a friction force $F = -F_{trap}$ (neglecting the very small hydrodynamic drag arising from the viscosity of the aqueous solution). Positive stage velocity is

defined as the microtubule moving with its plus end leading; in this case, the laser and the motor are moving toward the minus end. (B) Absolute value of the friction force as a function of drag speed for the same Kinesin molecule and microtubule shown in (A). Drag direction toward the microtubule's plus end (∇) and minus end (Δ) are indicated. (Schematic inset) Asymmetric potential landscape defining the periodicity δ, the asymmetry parameter Δ, the potential well depth U_0, and the forward k_+ and backward k_- rates. The asymmetry in the schematic is exaggerated to illustrate the asymmetry parameter. Adapted from Bormuth et al.[18]

CHAPTER 2

Observing the mushroom-to-brush transition for kinesins proteins

Bring out the tiger in you

Frosted Flakes

This Chapter was published in Langmuir as E.L.P. Dumont, H. Belmas, and H. Hess, *Observing the mushroom-to-brush transition for kinesin proteins,* **2013**, 29 (49), 15142-15145.

Objective and rationale

More than 30 years ago, Alexander and De Gennes derived scaling laws which showed that the height of grafted polymers is independent of grafting density at low densities ("mushroom" regime) and scales with the third root of the grafting density at high densities ("brush" regime).[13,56] The transition between the mushroom and the brush regime occurs when polymer chains start to overlap.[13,56,58,65] While studies frequently examine the scaling behavior in the brush regime,[66] few studies were able to observe the transition from mushroom to brush.[67,68] Wu et al., for example, experimentally observed the mushroom-to-brush transition for grafted polymers using polyacrylamides (average molecular weight: 17 kDa; radius of gyration: 5.4 nm).[59] As predicted, the height of polymers in the mushroom regime (18 nm) was constant and the height increased with an exponent of roughly 1/3 (0.37 – 0.4) in the brush regime. The transition from the mushroom to the brush regime occurred when the available area per chain occurred at a reduced grafting density $\Sigma = 6$, where $\Sigma = \sigma\pi R_g^2$, σ is the grafting density, and R_g is the radius of gyration of the polymer (**Figure 13**). However, the transition to a brush is gradual, with simulations showing an onset at $\Sigma = 1$[60] and a highly stretched brush regime reached in some cases at $\Sigma > 10$ due to polymer polydispersity and solvent effects.[57]

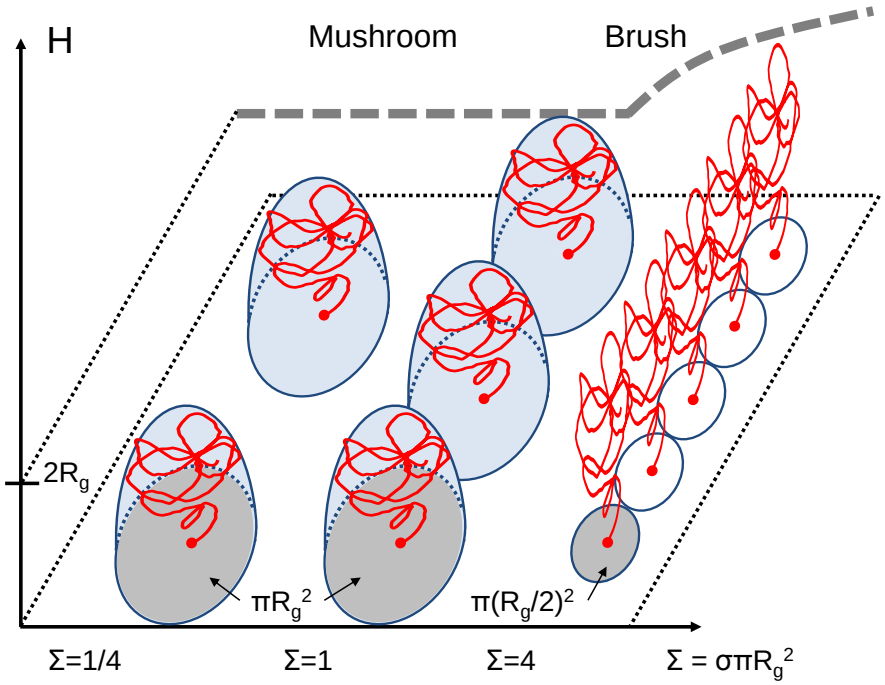

Figure 13. Schematic of the mushroom-to-brush transition. If the grafted polymer chains occupy approximately half-ellipsoids of radius R_g and height $2R_g$, the mushroom-to-brush transition is expected near a reduced grafting density of $\Sigma=1$,[60] while the height of polymers in the mushroom regime is predicted to be $2R_g$.[13]

The height of grafted polymer chains is not only of interest for synthetic coatings, but also in biological systems.[69,70] For example, microtubule-associated proteins (MAPs) maintain the spacing between microtubules by forming a dense brush,[71] which means that the height of the MAP brush plays a key physiological role in neural functioning. Here, we map out the mushroom-to-brush transition using kinesin-1 motor proteins,[72] which constitute a highly non-traditional polymer system. Kinesin-1 motor proteins are dimeric proteins which consist of a long tail which has evolved to attach to various intracellular cargos and a neck region connecting the two heads and the tail.[73] The 57 nm long protein consists of 7 segments of varying length (8, 15, 10, 5, 6, 8,

and 5 nm) which are connected by flexible regions.[74] It has been shown that by pre-coating a surface with casein proteins, which bind in a 2 nm thick monolayer, and supplementing the solution with additional casein, kinesins can be adhered to the surface by their tails only.[51,75] The casein layer effectively disrupts interactions between the kinesin head domains and the surface and prevents denaturation while maintaining motor activity. The height of surface-adhered kinesin-1 motors can be measured by letting fluorescently labeled microtubules (tubular assemblies of tubulin proteins with a diameter of 25 nm and a length of several micrometers) bind to the kinesins, and by measuring the elevation of the microtubules above the surface using fluorescence interference contrast (FLIC) microscopy.[74,76] FLIC microscopy is an interferometric method which utilizes the height-dependent efficiency of fluorescence excitation and emission near a reflecting surface to obtain height measurements with nanometer accuracy.[77-80] While the fluorescence intensity of an individual fluorophore above a reflecting surface is modulated in a non-linear fashion, the averaging over the fluorophores distributed over the cross-section of a microtubule results in a nearly linear relationship between microtubule brightness and height for heights between 15 and 60 nm above the surface in aqueous buffer (**Figure 14**).[74] Using a 12.5 nm silicon oxide layer as a spacer between the silicon and the solution, Kerssemakers et al.[74] showed that microtubules adhering to a layer of avidin proteins are positioned 3.5 ± 0.2 nm above the surface and that microtubules adhered to kinesins at a density of ~340 μm^{-2} are positioned 16.8 ± 1.9 nm above the silicon oxide surface. This height, measured for a reduced kinesin density of $\Sigma \approx 0.1$, is not significantly different from twice the Flory radius of gyration of kinesin proteins (9.5 nm) given by:[81]

$$R_g = \frac{1}{\sqrt{6}} \sqrt{\sum_{i=1}^{N} a_i^2}$$

Where the a_i are the lengths of the flexibly connected individual segments (8, 15, 10, 5, 6, 8, and 5 nm). The match between the kinesin height measured by Kerssemakers et al.[74] and twice the radius of gyration suggests that all the approximations and assumptions regarding the flexibility of the kinesin tail, the absence of kinesin-surface interactions, the presence of kinesin-microtubule interactions, and the short number of segments roughly cancel each other out to yield the mushroom height predicted by Alexander and deGennes.[13,56]

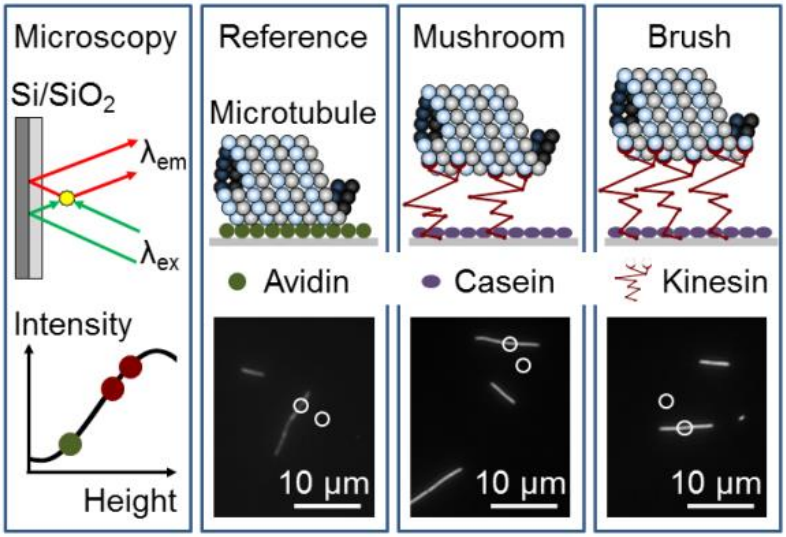

Figure 14. Experimental set-up. Interference between direct and reflected light leads to a modulation of the observed fluorescence intensity that is for fluorescently labeled microtubules linearly dependent on the distance to the surface between 20 and 60 nm.[74] The microtubule fluorescence intensity per length is determined by subtracting the integrated counts from a circular region adjacent to a microtubule from the integrated counts of a circular region on a microtubule. Microtubules adsorbed directly to an avidin surface served as a reference, while microtubules attached to kinesin showed a higher brightness which was dependent on kinesin surface density.

Experimental section

Kinesin brushes of varying density were prepared by adsorbing kinesins from solutions of different kinesin concentrations to a surface pre-coated with casein (See further sections). The kinesin stock solution concentration was determined from microtubule landing rate measurements with an error of 33% (see further sections).[82] By replacing microtubules with nanospheres in these landing rate experiments, it has been shown that the kinesin surface density is proportional to the initial kinesin solution concentration for kinesin surface densities between 10 μm^{-2} and 4,000 μm^{-2} (See further sections).[83] Microtubules were observed in a flow cell on the top surface of a silicon wafer coated with a silicon dioxide layer of 20 nm. The 20 nm silicon dioxide layer was used to place the microtubules at a distance from the reflecting silicon interface where the fluorescence intensity changes linearly with distance (**Figure 14**). The fluorescence intensity of microtubules on avidin was measured by flowing avidin proteins into the flow cell, washing with buffer to remove avidin proteins which did not adhere to the surface, flowing in microtubules, and flowing in an antifade solution to prevent photobleaching. When measuring the fluorescence intensity of microtubules on avidin, a high variance in the data was observed, corresponding to populations of microtubules aggregating with each other (as a result of positively charged avidin cross-linking negatively charged microtubules). A Bayesian clustering algorithm was used to separate these different populations and determine the average of fluorescence intensity of single microtubules (see further sections). This value was assumed to correspond to the previously determined height of 3.5±0.2 nm.[74] The fluorescence intensity of microtubules on kinesins was measured by flowing casein proteins into the flow cell to create a layer of adsorbed caseins and prevent kinesin proteins from collapsing on the surface,[84] flowing kinesin proteins into the flow cell at specific initial concentration, flowing microtubules into the flow cell, and finally flowing in an antifade solution to prevent photobleaching during imaging. The steps were 5 minutes separated in time, and Adenylyl-imidodiphosphate (AMP-PNP) was

used to inhibit kinesin activity and render bound microtubules stationary.[47] Two separately prepared batches of full-length kinesin-1 heavy chains were used, whose stock concentration was determined by landing rate experiments to be 23,800 ± 7,800 and 24,700 ± 8,000 μm^{-2}, respectively (See further sections). From each kinesin batch, six precise dilutions were added to individual flow cells. For each flow cell, 150 and 600 measurements of the fluorescence intensity of microtubules distributed over about ten fields of view using a 100x oil objective, a cooled CCD camera and an exposure time of 40 s, were taken. The standard deviation of the individual microtubule intensities for a given kinesin density was on the order of 10 % of the mean. Since the ratio of standard deviation to the mean of microtubule intensities on a glass surface (where a height dependent intensity modulation is absent) was similar (10%), this variation was mainly due to the distribution in the number of protofilaments in each microtubule (13, 14, 15 or 16).[85] Each data point in **Figure 15** represents the average of these measurements and the error bars a 95% confidence interval for the average. A detailed description of the methods can be found in the further sections.

Result and discussion

The interpretation of the kinesin-density dependent microtubule fluorescence is complicated by the fact that while the relative kinesin surface densities are well defined for each set of experiments, the uncertainty in the concentration of each stock solution is high (33% for both batches) and creates a proportional uncertainty in the absolute kinesin surface concentration. Therefore, the results for each batch are shown in separate graphs (**Figure 15a** and **Figure 15b**). In each graph, the fluorescence intensity measured for kinesin densities between 200 μm^{-2} and 1,500 μm^{-2} (the plateau characteristic for the mushroom regime) was averaged and identified with the height of 16.8 ± 1.9 nm measured by Kerssemakers et al. who utilized the identical protein, surface, and solution conditions.[74] The measurement of microtubule fluorescence

intensity on (1) avidin and (2) kinesin in the mushroom regime together with the known linear relationship between intensity and height enables the calibration of the microtubule intensity with respect to kinesin height (**Figure 15**).

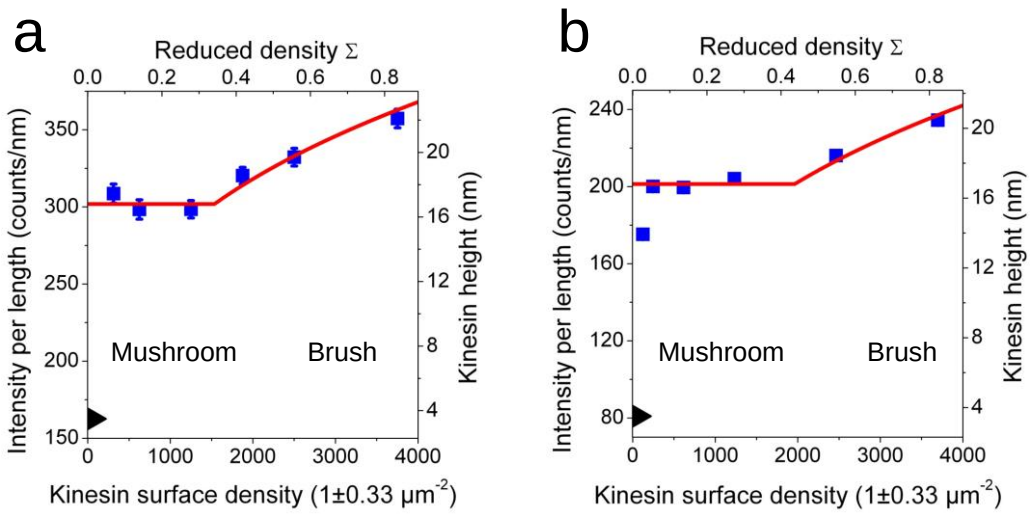

Figure 15. Mushroom-to-brush transition for kinesins. Fluorescence intensity of microtubules and kinesin height as a function of kinesin density for (a) batch A and (b) batch B of kinesin. The error bars on the data points represent a 95% confidence interval. The height in the brush regime is fit with $y = c*x^{1/3}$. The microtubule intensity on an avidin layer is indicated by the triangle. The absolute fluorescence intensity changes from Batch A to B are due to differences in the excitation intensity for the two experiments.

Taking Kerssemaker et al.'s experimental value of the mushroom height (H=16.8 nm) as corresponding to twice the radius of gyration, we would expect the mushroom-to-brush transition at roughly $4/(\pi H^2)$ = 4,500 μm^{-2} (equivalent to $\Sigma = \sigma\pi R_g^2 = 1$; **Figure 13**). We found a reduced density of $\Sigma = 0.34 \pm 0.11$ and $\Sigma = 0.43 \pm 0.14$ for Batches A and B respectively. Data points at kinesin densities of 1877, 2501, and 3752 μm^{-2} for Batch A and 2465 and 3698 μm^{-2} for Batch B show a marked increase in intensity, and the increase is consistent with a power law

scaling with the third root of the grafting density. The data point for Batch B at 123 μm^{-2} is an outlier compared to the data points at 247, 616 and 1,233 μm^{-2} which define the mushroom regime, which may reflect that at very low kinesin densities the microtubule begins to approach the surface due to the wide spacing of kinesins.

Compared with previous experimental observations of a mushroom-to-brush transition at reduced densities of $\Sigma = \sigma\pi R_g^2 = 1$ or higher,[57,59] here the mushroom-to-brush transition seems to occur relatively early ($\Sigma \approx 0.4$). The early onset may reflect the peculiarities of the grafted monodisperse kinesin protein, where two motor domains are tethered to a long stalk and repulsive surface–polymer interactions are minimized by the casein layer, or it may result from an underestimation of the kinesin footprint by modeling this semiflexible polymer with only seven segments as a long freely jointed chain unperturbed by interactions with the surface. Unfortunately, the relevant radius of gyration of kinesin in solution cannot be determined, because in the absence of another surface the tail binds to the motor domains and inhibits the motor activity.[73] While the biological purpose is presumably to conserve ATP if the motor is not attached to cargo by its tail, it also means that the radius of gyration of full-length kinesin which could be measured in solution is not representative of the space occupied by kinesin on the surface. Similarly, the measurement of the kinesin brush height with a second method is not straightforward. The tendency of proteins to non-specifically adsorb to surfaces would make atomic force microscopy measurements similar to those conducted by Yamamoto et al. [66] very challenging, and the measurement of the dried film thickness to calculate the brush height as conducted by Wu et al.[59] is impractical because the buffer solution contains salts and acids which precipitate during drying and contribute to the film height. Nevertheless, the change in the height of kinesin-1 brushes as a function of their surface density observed here is consistent with the theoretical framework of Alexander and De Gennes and is one of the few direct experimental observations of a mushroom-to-brush transition in the height of a grafted polymer brush.

Outlook

Kinesin-1 motors are also widely used in nanobiotechnology as components of molecular shuttle systems, where the end of the kinesin tail is non-specifically adsorbed to a glass or silicon surface coated with casein and the heads propel microtubules which serve as the motile, cargo-carrying elements.[86-88] In this context, the height of the surface-grafted kinesin-1 motors as a function of the grafting density is of great interest, because a changing conformation may have wide-ranging implications for cargo attachment, force generation and kinesin-microtubule interactions.[76] Finally, the mushroom-to-brush transition may affect the behavior of proteins with long, flexible domains in crowded cellular environments, for example when kinesin traffic along a microtubule becomes congested.[55] In this situation, the increasingly repulsive interactions between the tails may enhance the dissociation rate and prevent a traffic jam. Proteins with long coiled-coil regions separated by flexible hinges are also found on the Golgi and on endosomes where they collectively mediate vesicle fusion.[89] In these situations, the mushroom-to-brush transition may provide a physical switch which is exploited for intracellular decision making.

Materials and methods

Microtubules were polymerized by reconstituting a 20 µg aliquot of rhodamine-labeled, lyophilized tubulin (TL331M, Lot 357 from Cytoskeleton Inc, Denver, CO) with 6.25 µL polymerization buffer solution (BRB80 with 4 mM $MgCl_2$, 1 mM GTP, 5% dimethyl sulfoxide), and placing it at 37 °C for 30 minutes. The microtubules are then stabilized by diluting them a thousand-fold into BRB80 buffer (80 mM piperazine-N,N′ -bis(2-ethanesulfonic acid), 1 mM $MgCl_2$, 1 mM Ethylene Glycol Tetraacetic Acid, pH 6.9 with KOH) with 10 µM paclitaxel (Sigma, St Louis, MO). A kinesin construct consisting of the wild-type, full-length Drosophila melanogaster kinesin heavy chain and a C-terminal His-tag was expressed in Escherichia coli and purified using a Ni-NTA column.[73] Flow cells are constructed using one cover slip, one silicon wafer, and double-

sided tape as spacer. We used silicon wafers with a 20 nm oxide layer (Siliconsense Inc., 3'' diameter, <100>, SEMI std. flats, one side polished, prime grade, surface roughness < 2 Å, flatness < 9 μm). For microtubules adhering to avidin, a solution of 1 μM avidin (A2667, Invitrogen, City) in BRB80 buffer is flown into the flow cell. After 5 min, the solution is exchanged with BRB80 to wash out avidin which did not adsorb. After another five minutes, this solution is exchanged with microtubule solution containing an enzymatic antifade system[90] (16 nM tubulin, 10 μM Paclitaxel, 20 mM D-glucose, 20 μg mL^{-1} glucose oxidase, 8 μg mL^{-1} catalase, 10 mM dithiothreitol, and 1 mM Adenylyl Imidodiphosphate (AMP-PNP) in BRB80). AMP-PNP (Sigma, St Louis, MO) is an ATP analogue which arrests motor action.[47] After another five minutes, the solution in the flow cell is exchanged with a solution containing the enzymatic antifade system only (10 μM Paclitaxel, 20 mM D-glucose, 20 μg mL^{-1} glucose oxidase, 8 μg mL^{-1} catalase, 10 mM dithiothreitol in BRB80). For microtubules adhering to kinesins, a solution of 0.5 mg mL^{-1} casein in BRB80 buffer is flown into each flow cell. After 5 min, the solution is exchanged with the kinesin motor solution (kinesin, 0.5 mg mL^{-1} casein, 1 mM AMP-PNP in BRB80). After another five minutes, this solution is exchanged with microtubule solution containing an enzymatic antifade system[90] (16 nM tubulin, 0.5 mg mL^{-1} casein, 10 μM Paclitaxel, 20 mM D-glucose, 20 μg mL^{-1} glucose oxidase, 8 μg mL^{-1} catalase, 10 mM dithiothreitol, and 1 mM AMP-PNP in BRB80). After another five minutes, the solution in the flow cell is exchanged with a solution containing the enzymatic antifade system only (0.5 mg mL^{-1} casein, 10 μM Paclitaxel, 20 mM D-glucose, 20 μg mL^{-1} glucose oxidase, 8 μg mL^{-1} catalase, 10 mM dithiothreitol in BRB80). The openings of the flow cells are then sealed with a small amount of vacuum grease to prevent evaporation of the solutions. All experiments were performed at 24 °C.

Microtubules were imaged using a Nikon TE2000-U Epi-fluorescence microscope (Nikon, Melville, NY) equipped with an X-cite 120 lamp (EXFO, Ontario, Canada), an iXON DU885LC EMCCD camera (Andor, South Windsor, CT) and a 100x oil objective (NA 1.45).

For each flow cell, we took between 150 and 600 measurements of the fluorescence intensity of microtubules distributed over about ten fields of view using a 100x objective, a cooled CCD camera and an exposure time of 40 s.

Determination of the width of a microtubule footprint for use in landing rate measurements.

In previous landing rate measurements, the interaction area of the microtubule with the kinesin on the surface was assumed to be equal to the "footprint" of the microtubule given by the product of average microtubule length and width (25 nm).[82,91] This was tested by comparing the product of kinesin density and interaction area obtained from landing rate measurements with the kinesin density determined from a measurement of total protein concentration ($C = 0.243$ mg/mL – George Bachand, Center for Integrated Nanotechnology, Sandia National Laboratory, private communication) and a measurement of the relative kinesin content from gel densitometry ($f = 34\%$, gel provided by George Bachand). The implied width is given by:

$$w = \frac{2M\rho}{fCHL}$$

where $M = 220$ kDa is the molecular weight of kinesin-1 heavy chains (from the database www.uniprot.org #P17210), $H = 78 \pm 2.4$ µm is the height of a typical flow cell, $L = 2.18 \pm 0.125$ µm is the average length of microtubules used in the landing rate measurements, and $\rho = A\sigma = 375 \pm 68$ is the product of interaction area and kinesin density for the landing rate measurements shown in **Figure 16** and **Figure 17**.

Assuming a 5% uncertainty for the concentration C, we obtain $\boldsymbol{w = 20.0 \pm 5.0}$ **nm**, which is the value used for the width of the microtubule "footprint" in this work.

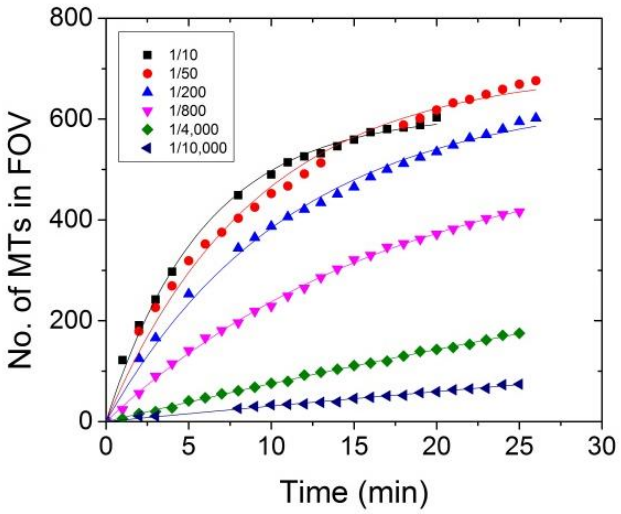

Figure 16. The number of microtubules (MTs) attached to the surface as function of time for the casein-coated glass exposed to kinesin solutions diluted from the stock solution. The field of view (FOV) was 80 µm × 80 µm. Experiment performed by Ashutosh Agarwal.

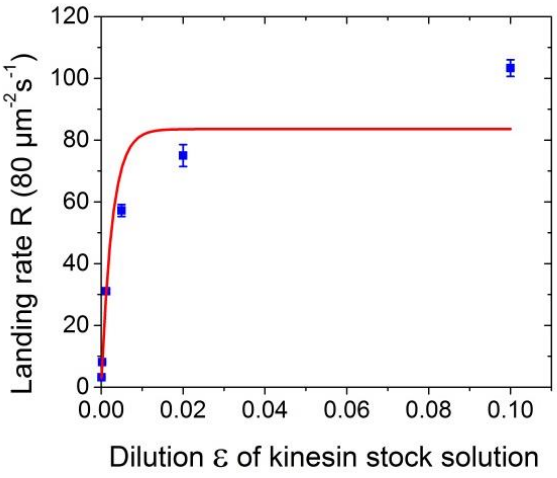

Figure 17. Landing rates *R* computed from the data shown in Figure 16 and plotted against the concentration of the kinesin solution for casein-coated glass surfaces. The equation $R = Z(1 - e^{-A\sigma\varepsilon})$ where *A* is the cross section area of a microtubule and σ is the kinesin grafting density corresponding to undiluted kinesin. We obtained $\rho = A\sigma = 375 \pm 68$. Experiment performed by Ashutosh Agarwal.

Measurement of kinesin grafting density with landing rate measurement.

To measure the kinesin grafting densities for batches A and B, we used a third batch C of kinesins to determine the average length of microtubules ($L = 2.7 \pm 0.2$ µm) and the diffusion limited landing rate Z. To do so, we measured the number of microtubules per field of view at several dilutions for the batch C (**Figure 18**). To each curve, we fit the equation $N = N_0\left(1 - e^{-\frac{R(t-t_{ini})}{N_0}}\right)$ where R, t_{ini} and N_0 are fit parameters. We then plot the landing rate R as a function of the dilution ε as shown in **Figure 19**. To this curve, we fit the equation $R = Z(1 - e^{-A\sigma\varepsilon})$ where $A = wL$ is the cross section area of a microtubule and σ is the kinesin grafting density corresponding to undiluted kinesin. We used $L = 2.7 \pm 0.2$ µm and $w = 20 \pm 5$ nm (determined in section (2) of further sections). The fit yields $\rho = A\sigma = 256 \pm 91$ and $Z = 1.09 \pm 0.27$ (80 µm)$^{-2}$s^{-1}. We then ran a landing rate experiment for batch A and B at 1/1000 dilution (**Figure 18**).

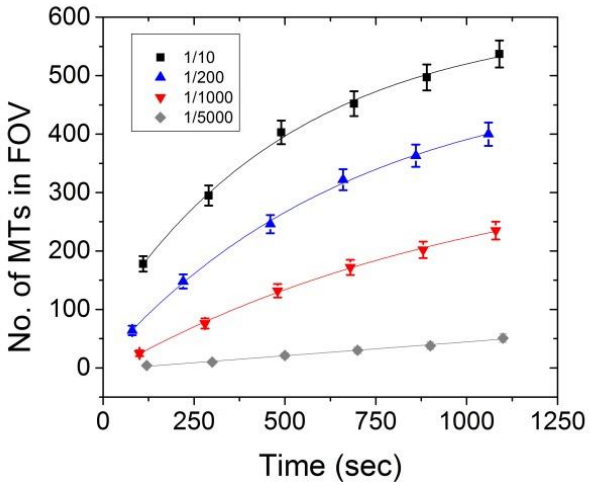

Figure 18. The number of microtubules (MTs) attached to the surface as function of time for the casein-coated glass exposed to kinesin solutions diluted from the batch C stock solution. The field of view (FOV) was 80 µm × 80 µm. Experiment performed by Ashutosh Agarwal.

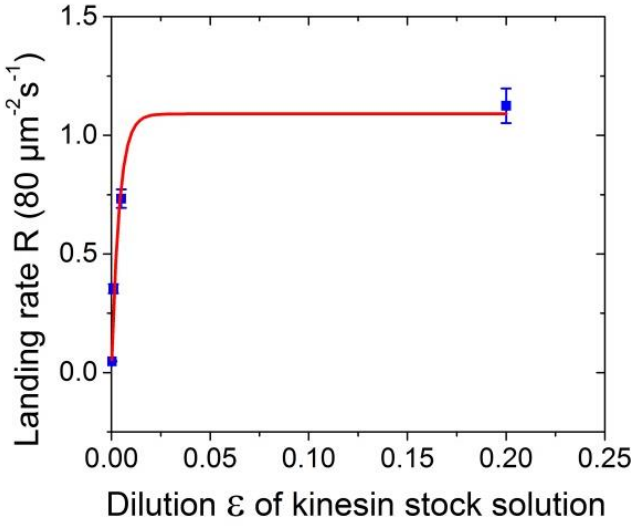

Figure 19. Landing rates R computed from the data shown in Figure 18 and plotted against the concentration of the kinesin solution for casein-coated glass surfaces. Experiment performed by Ashutosh Agarwal.

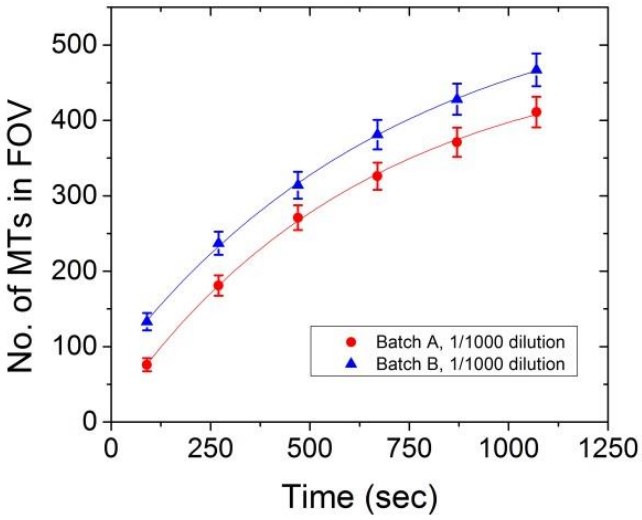

Figure 20. The number of microtubules (MTs) attached to the surface as function of time for the casein-coated glass exposed to kinesin solutions diluted from the batch A and B stock

solutions. The field of view (FOV) was 80 µm × 80 µm. Experiment performed by Ashutosh Agarwal.

Since the same microtubules were used for Batch A, B, and C and since the cross section area of a microtubule *A* and the diffusion limited landing rate *Z* were previously determined with Batch C, the kinesin surface densities which would be obtained from undiluted stocks for Batch A and B can be determined by fitting $N = N_0\left(1 - e^{-\frac{R(t-t_{ini})}{N_0}}\right)$ to their landing rate measurements (**Figure 20**), finding the fit value for *R*, and using the following equation:

$$\sigma = -\frac{1}{A\varepsilon}\ln\left(1 - \frac{R}{Z}\right)$$

The error on the kinesin densities is given by:

$$\left(\frac{\delta\sigma}{\sigma}\right)^2 = \left(\frac{\delta A}{A}\right)^2 + \left(\frac{\delta\varepsilon}{\varepsilon}\right)^2 + \frac{(\delta Z/Z)^2 + (\delta R/R)^2}{[\ln(1 - R/Z)]^2}$$

We found the kinesin density to be 23,761 ± 7,841 µm^{-2} for batch A and 24,651 ± 8,022 µm^{-2} for batch B as the kinesin densities which would be obtained from undiluted stock solution.

The kinesin grafting density does not saturate up to 4,000 µm^{-2}.

Agarwal et al. showed that the kinesin grafting density is proportional to the kinesin concentration in solution if the adsorption time is 5 min (Supplementary Figure 4 of Agarwal2012).[83]

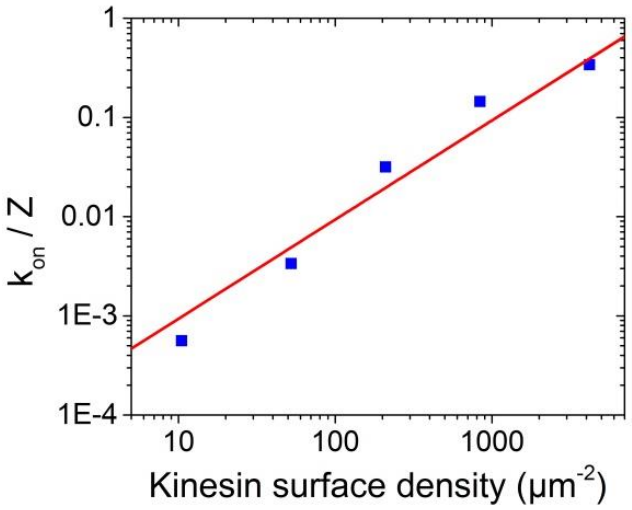

Figure 21. Relative attachment rate constants computed by dividing attachment rate constants k_{on} with the diffusion limited maximal landing rate Z determined on a bare glass surface. The error bars are roughly the size of the data points and represent the standard error. Adapted from Agarwal et al.[83]

Bayesian algorithm to separate normal distributions.

The fluorescence measurements of microtubules adhering to avidin are represented by N observations $y = (y_1, \dots, y_n)$. We assume that microtubules are either singles or aggregate to each other in doublets, triplets, etc. We designed a Bayesian clustering algorithm to differentiate these different populations. We assume that they are K populations of microtubules and that the fluorescence intensity of each population "k" follows a normal law of mean μ_k and standard deviation σ_k. We assume that a fluorescence measurement has the probability η_k to belong to the population "k". The probability density function f for the N observations y_n is given by:

$$\forall\, n \in [\![1, N]\!], \qquad f(y_n|v) = \sum_{k=1}^{K} \eta_k f_{\mathcal{N}_k}(y_n|\, \mu_k, \sigma_k^2)$$

Where $f_{\mathcal{N}_k}$ is the probability density function of a normal distribution of mean μ_k and standard deviation σ_k and $\nu = (\mu_1, \sigma_1^2, \eta_1, \dots, \mu_K, \sigma_K^2, \eta_K)$. We also have:

$$\forall\, k \in [\![1, K]\!],\ \eta_k \geq 0 \quad \text{and} \quad \sum_{k=1}^{K} \eta_k = 1$$

A common and convenient formulation introduces a latent unobserved random variable $S = (S_1, \dots, S_K)$ such as:

$$\forall\, k \in [\![1, K]\!], \qquad P(S_t = k) = \eta_k$$

The random variable S plays the role of an "indicator variable" and we can re-write the random variable Y that generates the observed data:

$$Y \sim \sum_{k=1}^{K} \mathbb{1}_{S_t = k}\, \mathcal{N}(\mu_k, \sigma_k^2)$$

Using Bayes' law, we obtain:

$$f(\nu, S \mid y) = f(y \mid \nu, S) \times f(S \mid \nu) \times f(\nu)$$

Where $f(\nu, S \mid y)$ is the distribution of posteriors, $f(y \mid \nu, S)$ is the likelihood function, $f(S \mid \nu)$ is the distribution of allocations conditional on knowing ν, and $f(\nu)$ is the distribution of priors. We can re-write the distribution of posteriors as:

$$f(\nu, S \mid y) \propto \left(\prod_{k=1}^{K} \prod_{n: S_n = k} \left(\frac{1}{\sigma_k^2} \right)^{\#_k/2} \exp\left\{ -\frac{1}{2} \sum_{n: S_n = k} \frac{(y_n - \mu_k)^2}{2\sigma_k^2} \right\} \right) \times \left(\prod_{k=1}^{K} \eta_k^{\#_k} \right) \times f(\nu)$$

Where $\#_k = \mathrm{Card}(\{n: S_n = k\})$. We then introduce the "prior" distributions of $f(\nu)$, that is μ_k, σ_k, and η:

$$\begin{cases} \forall\, k \in [\![1,K]\!]\ \ f(\mu_k|\sigma_k^2) \sim \mathcal{N}(a_0(k), A_0(k)\sigma_k^2) \\ \forall\, k \in [\![1,K]\!]\ \ f(\sigma_k^2) \sim \mathcal{G}^{-1}\big(b_0(k), B_0(k)\big) \\ f(\eta) \sim \mathcal{D}\big(c_0(1), \dots, c_0(K)\big) \end{cases}$$

Where $\mathcal{N}$ is a normal distribution, $\mathcal{G}^{-1}$ is an inverse-gamma distribution, and $\mathcal{D}$ is a Dirichlet distribution and where $a_0(k), A_0(k), b_0(k), B_0(k), c_0(k)$ are the prior parameters for $k \in [\![1,K]\!]$. The priors μ_k and σ_k are chosen to be dependent, a commonly used approach in Bayesian statistics because the "posterior" distributions $f(\mu_k, \sigma_k^2 \mid y)$ turn out to be in closed forms:

$$\begin{cases} \forall\, k \in [\![1,K]\!]\ \ f(\mu_k \mid \sigma_k^2, S, y) \sim \mathcal{N}(a_N(k), A_N(k)\sigma_k^2) \\ \forall\, k \in [\![1,K]\!]\ \ f(\,\sigma_k^2 \mid S, y) \sim \mathcal{G}^{-1}\big(b_N(k), B_N(k)\big) \\ f(\eta \mid S, y) \sim \mathcal{D}\big(c_N(1), \dots, c_N(K)\big) \\ \forall\, k \in [\![1,K]\!]\ \ \mathbb{P}(S_n = k \mid \nu, \mathrm{S}) = \mathbb{P}(S_n = k \mid \eta, \mathrm{S}) \propto \dfrac{1}{\sigma_k^2} \exp\left\{\dfrac{(y_n - \mu_k)^2}{2\sigma_k^2}\right\} \eta_k \end{cases}$$

And where the parameters $a_N(k), A_N(k), b_N(k), B_N(k), c_N(k)$ are the following:

$$\begin{cases} \forall\, k \in [\![1,K]\!]\ \ a_N(k) = \dfrac{A_0(k)}{\#_k A_0(k) + 1}\left(\dfrac{a_0(k)}{A_0(k)} + \#_k \bar{y}(k)\right) \\ \forall\, k \in [\![1,K]\!]\ \ A_N(k) = \dfrac{A_0(k)}{\#_k A_0(k) + 1} \\ \forall\, k \in [\![1,K]\!]\ \ b_N(k) = b_0(k) + \dfrac{\#_k}{2} \\ \forall\, k \in [\![1,K]\!]\ \ B_N(k) = B_0(k) + \dfrac{1}{2}\left(\#_k s_k^2(k) + \dfrac{\#_k / A_0(k)}{\#_k + A_0(k)}\big(\bar{y}(k) - b_0(k)\big)^2\right) \\ \forall\, k \in [\![1,K]\!]\ \ \bar{y}(k) = \dfrac{1}{\#_k} \displaystyle\sum_{n:S_n = k} y_n \\ \forall\, k \in [\![1,K]\!]\ \ s_k^2(k) = \dfrac{1}{\#_k} \displaystyle\sum_{n:S_n = k} \big(y_n - \bar{y}(k)\big)^2 \end{cases}$$

To draw the "posterior" distribution $f(\nu, S \mid y)$, we use Gibbs sampling, a Markov Chain Monte Carlo algorithm widely used in Bayesian inference.[92-94] The only requirement of Gibbs sampling is

to be able to draw from the conditional distributions $f(\mu_k, \sigma_k^2 \mid S, y)$, $f(\eta \mid S, y)$, and $f(S \mid \eta, y)$ for $k \in [\![1, K]\!]$ which we know. The Gibbs sampler we use is the following:

Initialize: Start with an initial classification $S^{(0)}$:

for $d = 1, \dots, D_0, \dots, D$

  Update the parameters $\left(a_N(k)\right)_d, \left(A_N(k)\right)_d, \left(b_N(k)\right)_d, \left(B_N(k)\right)_d, \left(c_N(k)\right)_d$ using $S^{(d-1)}$

  Draw $\eta^{(d)}$ from $f\left(\eta^{(d)} \mid S^{(d-1)}, y\right) \sim \mathcal{D}\left(c_0(1), \dots, c_0(K)\right)$

  for k = 1,…, K

    Draw $\sigma_k^{2(d)}$ from $f\left(\sigma_k^{2(d)} \mid S^{(d-1)}, y\right) \sim \mathcal{G}^{-1}\left(b_N(k), B_N(k)\right)$

    Draw $\mu_k{}^{(d)}$ from $f\left(\mu_k{}^{(d)} \mid \sigma_k^{2(d)},\ S^{(d-1)},\ y\right) \sim \mathcal{N}\left(a_N(k), A_N(k)\,\sigma_k^{2(d)}\right)$

  End

  Update the probabilities $p\left(S^{(d)} = k \mid \eta^{(d)}, \mathrm{y}\right)$

  Draw a new classification $S^{(d)}$ using $p\left(S^{(d)} = k \mid \eta^{(d)}, \mathrm{y}\right)$

End

Drop the D_0 draws

Result: D draws $(\nu, S)^{(D_0+1)}, \dots, (\nu, S)^{(D)}$

We chose to iterate the algorithm five times to obtain stable results. In the first iteration, we use prior parameters drawn from data points:

$$\begin{cases} \forall\, k \in [\![1, K]\!],\ a_0(k) = \bar{y} = \dfrac{1}{N}\displaystyle\sum_{n=1}^{N} y_n \\ \forall\, k \in [\![1, K]\!],\ A_0(k) = 1 \\ \forall\, k \in [\![1, K]\!], b_0(k) = 2 \\ \forall\, k \in [\![1, K]\!], B_0(k) = \dfrac{1}{N}\displaystyle\sum_{n=1}^{N} (y_n - \bar{y})^2 \\ \forall\, n \in [\![1, K]\!], c_0(k) = 1 \end{cases}$$

Each iteration leads to a set of posterior parameters and D drawings from the posterior distributions of $\hat{\mu}_k$, $\hat{\sigma}_k^2$, $\hat{S}$, and $\hat{\eta}_k$. We use these drawings to update the priors of the next iteration such as:

$$\begin{cases} \forall\, k \in [\![1, K]\!],\ a_0(k) = \dfrac{1}{D}\displaystyle\sum_{d=1}^{D} \hat{\mu}_k^{(d)} \\ \forall\, k \in [\![1, K]\!],\ A_0(k) = 1 \\ \forall\, k \in [\![1, K]\!], b_0(k) = 2 \\ \forall\, k \in [\![1, K]\!], B_0(k) = \dfrac{1}{D}\displaystyle\sum_{d=1}^{D} \hat{\sigma}_k^{(d)} \\ \forall\, k \in [\![1, K]\!],\ c_0(n) = \dfrac{1}{D}\displaystyle\sum_{d=1}^{D} \hat{\eta}_k^{(d)} \end{cases}$$

This algorithm leads to the results shown in the next section.

Use of the algorithm to separate populations of microtubules adhering to avidin.

Taking the algorithm described in the previous section with *K* = 5, we ran the algorithm five times. If the algorithm only finds 4 Gaussians, we would re-run five times the algorithm with *K* = 4, etc.

For batch A, the algorithm found 3 normal distributions (**Figure 22**). For the fluorescence of single microtubules adhering to avidin, the average of population 1 (blue squares) is 162.6 ± 4.0 counts/nm. For batch B, the algorithm found 4 normal distributions as seen below (**Figure 22**). For the fluorescence of single microtubules adhering to avidin, the average of population 1 (blue squares) is 80.8 ± 1.5 counts/nm.

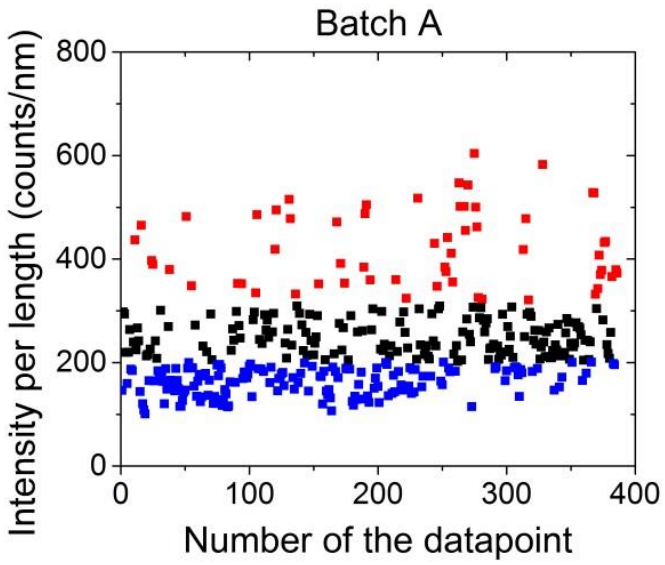

Figure 22. Results of the Bayesian algorithm for microtubules adhering to avidin from Batch A. Blue squares are population 1, black squares are population 2, red squares are population 3.

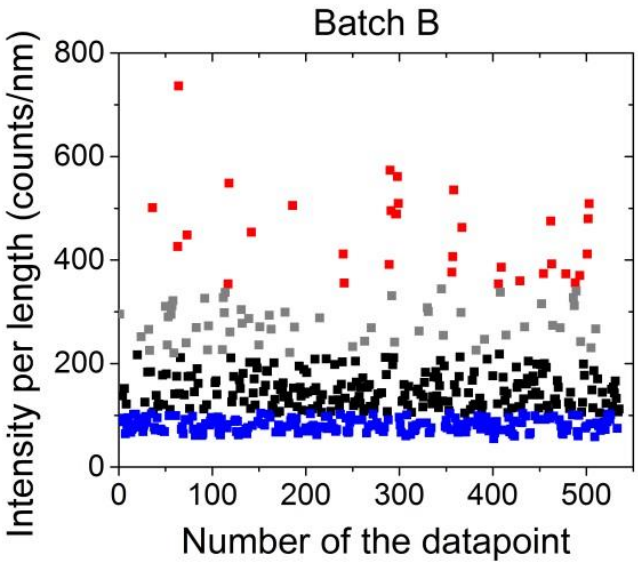

Figure 23. Results of the Bayesian algorithm for microtubules adhering to avidin from Batch B. Blue squares are population 1, black squares are population 2, grey squares are population 3, red squares are population 4.

MATLAB code of the Bayesian algorithm

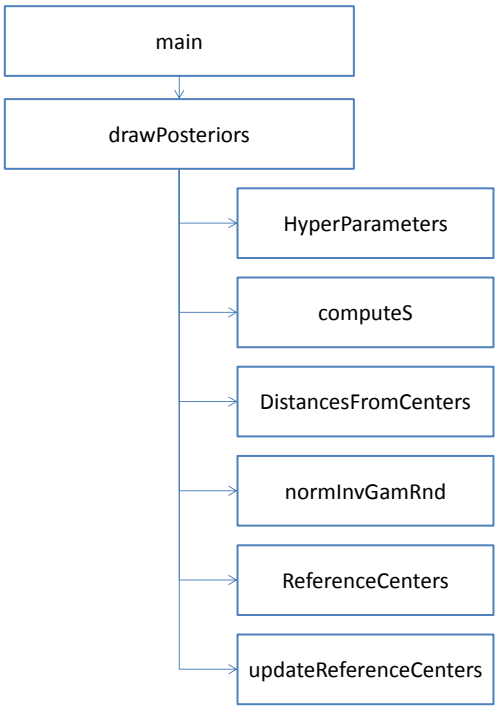

Figure 24. Structure of the MATLAB code for the Bayesian algorithm.

"main.m"

```
%% Initialization

clear all
close all

% Need a file called "data", a vector of all the data to be analyzed
load('data_avidin.mat')  ;
%data             =               data_avidin.d120808 ;
data             =               data_avidin.d130128 ;

%Parameters
N                =               10000            ; % Draws
K_true           =               3                ; % Useless
M                =               100              ; % Window size
N_0              =               1000             ; % Burning period


%% Priors

K                =               4                ; % Nb of Gaussians


% Do not change "A", "b"
A                =               ones(1,K)        ;
b                =               4.*ones(1,K)     ;

% Update "a" with mean(postMu) after running the algorithm
a                =               [ 1.0 ...        % Gaussians means
                                   1.0 ...
                                   1.0 ...
                                   1.0]            ;
% Update "B" with mean(postSigma2) after running the algorithm
B                =               [ 10  ...
                                   10  ...
```

```
                                    10  ...
                                    10 ]            ;
% Update "c" with mean(postS) after running the algorithm
c                   =               [ 0.25 ...
                                      0.25 ...
                                      0.25 ...
                                      0.25]          ; %  proportions of Gaussians
                                                       %  The sum must be
                                                       %  equal to 1

%% Display data in the form of histograms
figure(1);

[n xout]            =               hist(data,100)    ;
n                   =               n./sum(n)         ;

bar(xout,n);
clear xout n;
hold on

%% Draws

[postMu, postSigma2, postEta, postS, modesMeans, modesVariances] = ...
     drawPosteriors(data,N,M,N_0,K,a,A,b,B,c,1);


figure(3)
plot(postMu);
```

"drawPosteriors.m"

```
function [ postMu, postSigma2, postEta, postS, modesMeans, modesVariances ...
         ] = drawPosteriors( data, N, M, N_0, K, a, A, b, B, c, inLinePermutation )


%% Initialize parameters
T                   =               size(data,1)      ;
postMu              =               zeros(N,K)        ;
postSigma2          =               zeros(N,K)        ;
postEta             =               zeros(N,K)        ;
postS               =               zeros(T,K)        ;
gammaDraw           =               zeros(1,K)        ;

%% Temporary outputs from MCMC draws
tempMu              =               zeros(1,K)        ;
tempSigma2          =               zeros(1,K)        ;
tempEta             =               zeros(1,K)        ;

%Vector that store data about location/scale of each mode (one line per
%estimated parameter
modesMeans          =               zeros(2*K, factorial(K)) ;
modesVariances      =               zeros(2*K, factorial(K)) ;

cancelled           =               0                 ;
isBurning           =               1                 ;

%% Preliminary classification S_0 : draw from a uniform distribution
S                   =               randi(K,T,1)      ;

%% Initialize the waitbar
h                   =               waitbar(0,'Burning period...', ...
                                            'CreateCancelBtn', ...
                                            'setappdata(gcbf,''canceling'',1)');
setappdata(h,'canceling',0)


%% Burning period (1000 points) + sample of M points for reference centers
for m = 1:(M + N_0)

    % Check for Cancel button press
    if getappdata(h,'canceling')
        cancelled = 1;
        break
    end

    if m > N_0
        isBurning = 0;
        if inLinePermutation == 1
            waitbar( m / (M + N_0), h, 'Initial draws...');
        else
            waitbar( m / (M + N_0), h, 'Drawing from posterior...');
        end
    end

    %Update hyperparameters
    [ aT AT bT BT cT ] = HyperParameters( data, K, a, A, b, B, c, S  );
```

```
    %Draw (mu_i, sigma2_i)
    for k = 1:K
        [ mu sigma2 ]  = normInvGamRnd(1, aT(k), AT(k), bT(k), BT(k));

        %Store draws in temp variables
        tempMu(k)     = mu     ;
        tempSigma2(k) = sigma2 ;

    end

    clear k mu sigma2

    %Store values in final vectors only after burning period
    if ~isBurning
        postMu(m-N_0,:)     = tempMu     ;
        postSigma2(m-N_0,:) = tempSigma2 ;
    end

    %Draw eta
    for k = 1:K
        gammaDraw(k)        = gamrnd(cT(k),1) ;
    end

    %Store in temp variable
    tempEta                 = gammaDraw ./ sum(gammaDraw) ;

    %Store values only after burning period
    if ~isBurning
        postEta(m-N_0,:)    = tempEta ;
    end

    %Update distribution of S
    for k = 1:K
        postS(:,k) = 1/(sqrt(2*pi*tempSigma2(k))) .* ...
                     exp( - ((data - tempMu(k)).^2) ./ ...
                     (2*tempSigma2(k)) ) .* tempEta(k);
    end

    for t = 1:T
        postS(t,:) = postS(t,:) ./ sum(postS(t,:)) ;
    end

    %Draw S
    S              = computeS(T, postS) ;

    %Update waitbar
    waitbar(m / (M + N_0) ,h);
end

%% Delete temporary varaibles
clear tempMu tempSigma2 tempEta m isBurning;

figure(2)
subplot(2,1,1)
plot(postMu((1:100),:));
subplot(2,1,2)
plot(postSigma2((1:100),:));

%% Compute reference centers
if inLinePermutation == 1
    [modesMeans modesVariances permutations] = ReferenceCenters(K, ...
                                               postMu(1:M,:), postSigma2(1:M,:) );
end


%% Reset the waitbar
waitbar(0,h,'Drawing from posterior...');

for n = M+N_0+1 : N_0+N

    % Check for Cancel button press
    if or(getappdata(h,'canceling'), cancelled)
        break
    end

    %Update hyperparameters
    [ aT AT bT BT cT ] = HyperParameters(data,K, a, A, b, B, c, S  );

    %Draw (mu_i, sigma2_i)
    for k = 1:K
        [ mu sigma2 ] = normInvGamRnd(1, aT(k), AT(k), bT(k), BT(k));
        postMu(n-N_0,k) = mu;
        postSigma2(n-N_0,k) = sigma2;
    end

    clear k mu sigma2

    if inLinePermutation == 1
        %Compute closest reference center
        indexMinDistance = DistancesFromCenters(K, postMu(n-N_0,:), ...
                           postSigma2(n-N_0,:), modesMeans, modesVariances);
```

```
        %Relabel output so that the closest reference center is the first one
        if indexMinDistance ~= 1

            %Get permutation
            [order invPerm] = sort(permutations(indexMinDistance,:)) ;
            perm            = invPerm(permutations(1,:))             ;

            %Permute draws
            postMu(n-N_0,:) = postMu(n-N_0, perm);
            postSigma2(n-N_0,:) = postSigma2(n-N_0, perm);

            clear order invPerm perm
        end

        %Update modes
        [modesMeans modesVariances] = updateReferenceCenters(n, N_0, K,...
                                      modesMeans, modesVariances, ...
                                      postMu(n-N_0,:), postSigma2(n-N_0,:), ...
                                      permutations);
    end

    %Draw eta
    for k = 1:K
        gammaDraw(k) = gamrnd(cT(k),1);
    end

    postEta(n-N_0,:) = gammaDraw ./ sum(gammaDraw);

    %Update distribution of S
    for k = 1:K
        postS(:,k) = 1/(sqrt(2*pi*postSigma2(n-N_0,k))) .* ...
                     exp( - ((data - postMu(n-N_0,k)).^2) ./ ...
                     (2*postSigma2(n-N_0,k)) ) .* postEta(n-N_0,k);
    end

    for t = 1:T
        postS(t,:) = postS(t,:) ./ sum(postS(t,:));
    end

    %Draw S
    S = computeS(T, postS);

    waitbar((n-M-N_0) / N,h);
end

clear n S gammaDraw

delete(h)

close(figure(1));

end
```

"HyperParameters.m"

```
function [ aT AT bT BT cT ] = HyperParameters( data, K, a, A, b, B, c, S  )

%Compute hyper parameters for a mixture of K gaussians:
% y_t = p_1 * (mu1 + sigma1*n1_t) +...+ p_K * (muK + sigma2*nK_t) with:
% nk ~ N(muk, sigma1^2) with (muk, sigmak^2) ~ NIG(akT, AkT, bkT, BkT)
% S_t = ind(multnom(p)) with p ~ D(c_1,...,c_k)

%Initialize parameters
count          =   zeros(1,K)   ;
aT             =   zeros(1,K)   ;
AT             =   zeros(1,K)   ;
bT             =   zeros(1,K)   ;
BT             =   zeros(1,K)   ;
cT = zeros(1,K);

for k = 1:K
    count(k)   =   sum(S==k)    ;

    if count(k) > 0
        cT(k) =   c(k) + count(k)  ;
        bT(k) =   b(k) + count(k)  ;
        BT(k) =   B(k) + count(k)*var(data(S==k)) + (count(k)/A(k)) / ...
                  (count(k) + 1/A(k)) * ( (mean(data(S==k) - a(k))).^2 );
        AT(k) =   1/(count(k) + 1/A(k));
        aT(k) =   (1/A(k))/( count(k) + 1/A(k) ) * a(k) + count(k)/( ...
                  count(k) + 1/A(k))*mean(data(S==k));
    else
        cT(k) =   c(k);
        bT(k) =   b(k);
        BT(k) =   B(k);
```

```
        AT(k) =   A(k);
        aT(k) =   a(k);
    end

end

end
```

"normInvGamRnd.m"

```
function [ x1 x2 ] = normInvGamRnd( N, a, A, b, B )
%Draws N values from the normal inverse gamma distribution NIG(a,A,b,B)
% x1 | x2 ~ N(a, A*x2)
% x2 ~ IG( b/2, B/2)

%% Draw x2 ~ IG(b/2, B/2)
alpha     =       b/2                 ;
beta      =       B/2                 ;
y         =       gamrnd(alpha,1,N,1) ; %y ~ Gamma(alpha, 1)
x2        =       beta ./ y           ; %beta / y ~ IG(alpha, beta)


%% Draw x1 | x2 ~ N(a, A*x2)
mu        =       a                   ;
sigma     =       sqrt(x2.*A)         ;
x1        =       normrnd(mu, sigma)  ;

end
```

"computeS.m"

```
function [ S ] = computeS( T, probs )

S = zeros(T,1);

for t = 1:T
    mnDraw = mnrnd(1, probs(t,:));

    indice = 0;

    for i = 1:size(probs,1)
        if mnDraw(i) == 1
            indice = i;
            break
        end
    end

    S(t) = indice;
end

end
```

"DistancesFromCenters.m"

```
function [ indexMinDistance ] = DistancesFromCenters( K, currentMus, ...
                                 currentSigma2s, modesMeans, modesVariances )

currentCenter            =   zeros(2*K, 1)      ;

for k = 1:K
    currentCenter(2*k-1) =   currentMus(k)      ;
    currentCenter(2*k)   =   currentSigma2s(k)  ;
end

distances                =   sum( ( ( modesMeans - repmat(currentCenter, 1, ...
                             factorial(K)) ).^2 ) ./ modesVariances, 1 );


[minDistance indexMinDistance] = min(distances);

clear minDistance distances;

end
```

"ReferenceCenters.m"

```
function [ modesMeans modesVariances permutations ] = ReferenceCenters( K, postMu, postSigma2 )

M                       =   size(postMu,1)          ;
modesMeans              =   zeros(2*K, factorial(K)) ;
modesVariances          =   zeros(2*K, factorial(K)) ;

%Compute means
initialParamsMeans      =   zeros(2,K)              ;
initialParamsVariances  =   zeros(2,K)              ;

%% Average the first M draws (after burning period) to get
for k = 1:K
    initialParamsMeans(:,k)     = [ mean(postMu((1:M),k)); ...
                                    mean(postSigma2((1:M),k)); ];
    initialParamsVariances(:,k) = [ var((postMu((1:M),k)),1); ...
                                    var((postSigma2((1:M),k)),1); ];
end

%Generate all permutations for mu and sigma2 (both for means and variances
%of estimates)
permutations            =   perms( (1:K) )          ;
[ orders invPerms ]     =   sort(permutations,2)    ;

for p = 1:factorial(K)
    permutations(p,:)   =   permutations(p,invPerms(1,:)) ;
end

%% Build matrices that contained all possible permuted values
for k = 1:factorial(K)

    mus                 =   initialParamsMeans(1,:)     ;
    sigma2s             =   initialParamsMeans(2,:)     ;
    varMus              =   initialParamsVariances(1,:) ;
    varSigma2s          =   initialParamsVariances(2,:) ;

    muPerms             =   mus(permutations)           ;
    varOfMuPerms        =   sigma2s(permutations)       ;

    sigma2Perms         =   varMus(permutations)        ;
    varOfSigma2Perms    =   varSigma2s(permutations)    ;

end

%Clear temp variables
clear k mus sigma2s varMus varSigma2s orders;

%% Store every mode
for m = 1:factorial(K)

    for k = 1:K

        %Store location data and scale data in separated matrices
        modesMeans(2*k-1,m)       =   muPerms(m,k)        ; %Mean of mu
        modesVariances(2*k-1,m)   =   varOfMuPerms(m,k)   ; %Variance of mu
        modesMeans(2*k,m)         =   sigma2Perms(m,k)    ; %Mean of sigma2
        modesVariances(2*k,m)     =   varOfSigma2Perms(m,k); %Variance of sigma2
    end
end

clear k initialParamsMeans initialParamsVariances muPerms sigma2Perms ...
      varOfMuPerms varOfSigma2Perms;


end
```

"updateReferenceCenters.m"

```
function [ modesMeans modesVariances ] = updateReferenceCenters( n, N_0,...
          K, modesMeans, modesVariances, postMu, postSigma2, permutations )


%% Update first center
meanOfMus       =               zeros(1,K)      ;
meanOfSigma2s   =               zeros(1,K)      ;
varOfMus        =               zeros(1,K)      ;
varOfSigma2s    =               zeros(1,K)      ;

for k = 1:K
    %Save current modes' means
    prevMeansOfFirstMode = modesMeans(:,1);

    %Update modes means
```

```
    modesMeans(2*k-1,1) = (n-N_0-1)/(n-N_0)*modesMeans(2*k-1,1) + ...
                          1/(n-N_0)*postMu(k); %Means of Mus
    meanOfMus(k)        = modesMeans(2*k-1,1);

    modesMeans(2*k,1)   = (n-N_0-1)/(n-N_0)*modesMeans(2*k,1) + ...
                          1/(n-N_0)*postSigma2(k); %Means of sigma2s
    meanOfSigma2s(k)    = modesMeans(2*k,1);

    %Update modes variances
    modesVariances(2*k-1,1) = (n-N_0-1)/(n-N_0)*modesVariances(2*k-1,1) ...
                              + (n-N_0-1)/(n-N_0)*( ...
                              (prevMeansOfFirstMode(2*k-1) - ...
                              modesMeans(2*k-1,1)).^2 ) + 1/(n-N_0)*( ...
                              (postMu(k) - modesMeans(2*k-1,1)).^2 ); %Variances of mus
    varOfMus(k)             =  modesVariances(2*k-1,1);

    modesVariances(2*k,1)   = (n-N_0-1)/(n-N_0)*modesVariances(2*k,1) + ...
                              (n-N_0-1)/(n-N_0)*( ...
                              (prevMeansOfFirstMode(2*k) - ...
                              modesMeans(2*k,1)).^2 ) + 1/(n-N_0)*( ...
                              (postSigma2(k) - modesMeans(2*k,1)).^2 ); %Variances of sigma2s
        varOfSigma2s(k)     = modesVariances(2*k,1);
end

clear prevMeansOfFirstMode k

%% Obtains other centers by permutation

[order invFirstPerm] =            sort(permutations(1,:))     ;
meanOfMus            =            meanOfMus(invFirstPerm)     ;
meanOfSigma2s        =            meanOfSigma2s(invFirstPerm) ;
varOfMus             =            varOfMus(invFirstPerm)      ;
varOfSigma2s         =            varOfSigma2s(invFirstPerm)  ;

clear order

for m = 2:factorial(K)

    %Permute parameters individually
    permutedMus      =            meanOfMus(permutations(m,:))     ;
    permutedSigma2s  =            meanOfSigma2s(permutations(m,:)) ;
    permutedVarOfMus =            varOfMus(permutations(m,:))      ;
    permutedVarOfSigma2s =        varOfSigma2s(permutations(m,:))  ;

    %Store them into modes' matrix
    for k = 1:K
        modesMeans(2*k-1,m)     = permutedMus(k)          ;
        modesMeans(2*k,m)       = permutedSigma2s(k)      ;
        modesVariances(2*k-1,m) = permutedVarOfMus(k)     ;
        modesVariances(2*k,m)   = permutedVarOfSigma2s(k) ;
    end
end

%% Clear temp variables
clear m k permutedMus permutedSigma2s permutedVarOfMus ...
      permutedVarOfSigma2s meanOfMus meanOfSigma2s varOfMus varOfSigma2s

end
```

CHAPTER 3

Molecular wear of microtubules propelled by surface-adhered kinesins

A morning of awkwardness is far better than a night of loneliness

Hank Moody in Californication

This Chapter is currently under review by Nature Nanotechnology as E.L.P. Dumont and H. Hess, *Molecular wear of microtubules propelled by surface-adhered kinesins.*

Objective and rationale

Wear, the progressive loss of material from a body caused by contact and relative movement, is a major concern not only in engineering but also in biology.[8,10,14,15] Advances in nanotechnology both enable the study of the origins of wear processes at the atomic and molecular scale and demand the prediction and control of wear in nanoscale systems.[11,16,17] Here, we show that wear occurs in an in vitro system consisting of microtubules gliding across a surface coated with kinesin-1 motor proteins, and that energetic considerations suggest a molecule-by-molecule removal of tubulin proteins. The wear rates show a complex dependence on sliding velocity and kinesin density, which – in contrast to the friction behavior between microtubules and kinesin[18] – cannot be explained by simple chemical reaction kinetics.

Biomolecular systems distinguish themselves by the abundance of active movement on the nanoscale, which is enabled by the transduction of chemical energy into mechanical work by polymerization processes and motor proteins.[39] The active movement is accompanied by dissipative processes which can be conceptually understood as "protein friction".[18,95] Here, we explore if active movement of microtubules gliding on kinesins also leads to "protein wear". Wear is a fundamentally distinct process from breaking or targeted disassembly. Targeted disassembly of microtubules, for example, is performed by specialized motor proteins in order to regulate the size of the mitotic spindle.[41] During disassembly, mechanical stresses are generated for the very purpose of removing material. In contrast, wear is an undesired result of stresses caused by the normal operation of a mechanical system leading ultimately to failure of the system.[14] Within living systems, the effects of wear are often mitigated and obscured by the constant turnover of building blocks. Here, an *in vitro* system is employed where nanoscale movement and its consequences can be studied in the absence of restorative mechanisms.

Experimental design

In our experimental system known as the gliding motility assay (**Figure 25a**),[39] kinesin-1 proteins are non-specifically adhered to a surface at a defined density and propel microtubules across the surface at velocities up to 1 μm s^{-1} controlled by the ATP concentration. Microtubules, cytoskeletal filaments with a diameter of 25 nm, are polymerized from dimers of rhodamine-labeled tubulin protein into tubular filaments with a length of a few micrometers and stabilized against depolymerization by the addition of paclitaxel (taxol).[49,50] At the kinesin densities employed here, kinesins bind to a microtubule with an average spacing of as little as 10 nm and perform about a hundred 8 nm steps along one of its 12-16 protofilaments before they unbind.[39] As a result, a gliding microtubule experiences the stresses associated with up to 1 million individual kinesin steps per minute.

Microtubules – imaged by fluorescence microscopy – glide smoothly in a persistent random walk across the surface (**Figure 25b,c**) unless they encounter a defective kinesin.[96] Except for rare events where an entire segment of a microtubule breaks off, microtubules do not exhibit visible signs of degradation during smooth gliding on a timescale of minutes.[52] Here, if a microtubule experiences a breaking event (which occur at a rate below 10^{-3} s^{-1}) it is not included in the analysis. However, measurements of the length of smoothly gliding microtubules with subpixel accuracy, enabled by the FIESTA filament tracking software,[53] reveal a shortening of each microtubule at rates between 0 and 1 nm s^{-1} (**Figure 25d** and **Figure 26a**). For these measurements, images are acquired every 5 s with an exposure time of 200 ms.

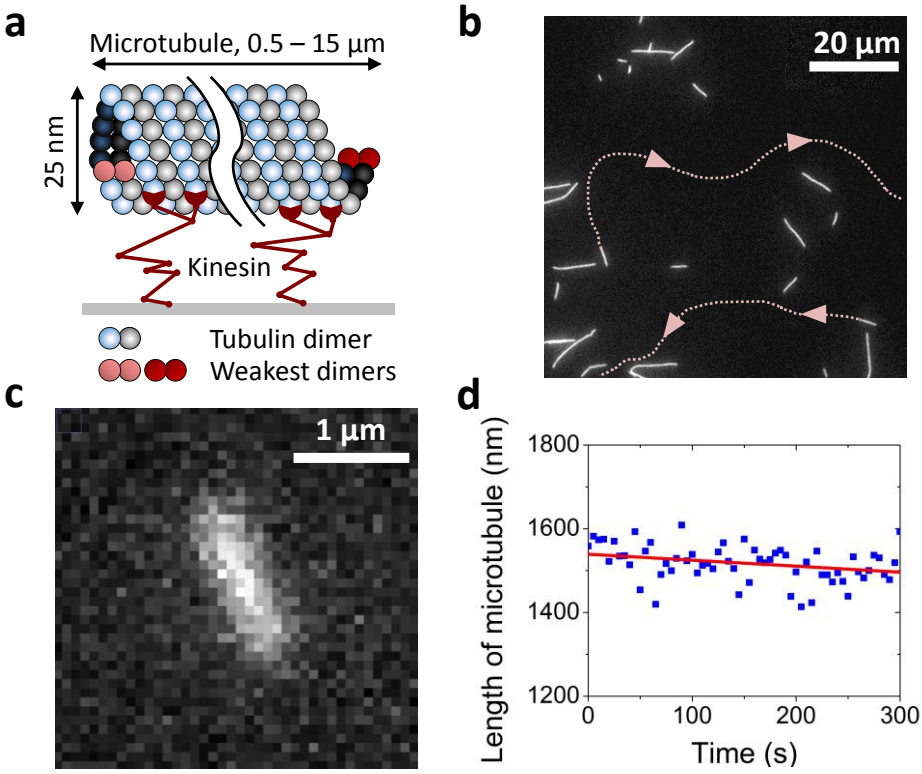

Figure 25. Length measurements of gliding microtubules. (a) Kinesin proteins spaced between 10 and 100 nm apart propel microtubules with 8 nm steps at ATP-dependent velocities up to 1 µm s^{-1}. (b) Traces of two microtubules identified by the filament tracking software. (c) Diffraction-limited fluorescence microscopy image of a rhodamine-labeled microtubule. (d) Length of the microtubule shown in (c) over time calculated by the filament tracking software over time. A linear fit determines a shrinking rate with an error of less than 0.7 nm s^{-1}.

Results

When microtubules are not gliding, which is obtained by replacing the ATP with the non-hydrolyzable ATP analogue AMP-PNP,[47] the shrinking rates of individual microtubules vary between -0.4 nm s^{-1} and 0.6 nm s^{-1} independent of kinesin density (**Figure 26a**). This distribution originates primarily from the length measurement error (See further sections). The average

shrinking rate of stationary microtubules is 0.068 ± 0.005 nm s^{-1} which reflects that microtubules tethered by kinesins to a surface depolymerize despite taxol-stabilization (**Figure 26a**) and is in quantitative agreement with previous measurements.[97] At a kinesin density of 2,500 ± 800 μm^{-2} and a gliding velocity of 199 ± 3 nm s^{-1}, the average shrinking rate is increased significantly to 0.36 ± 0.04 nm s^{-1}. Therefore, microtubule gliding causes wear ($p<10^{-11}$).

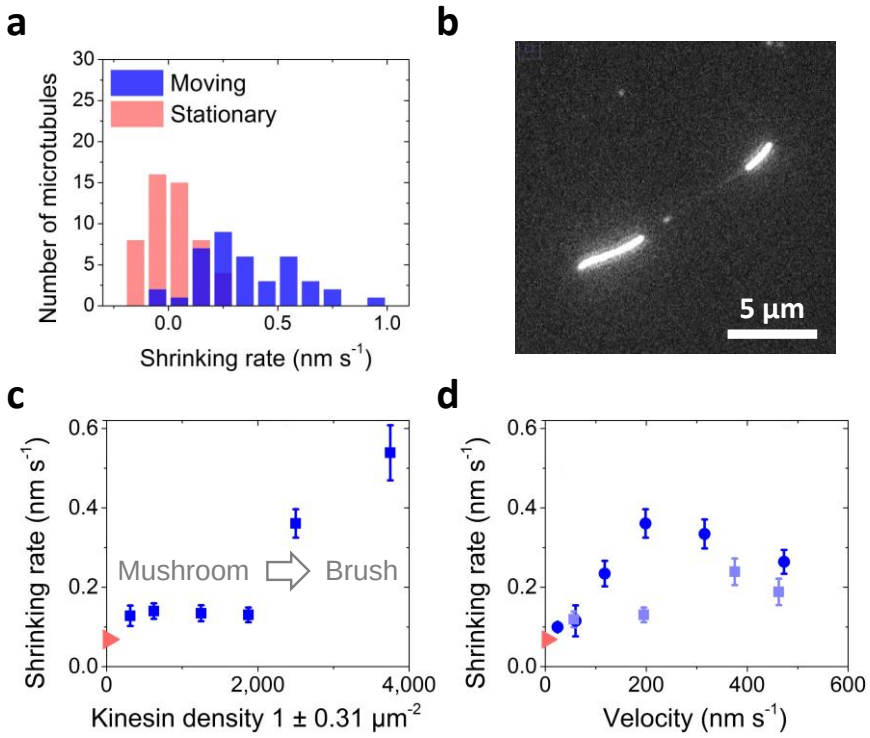

Figure 26. Microtubule shrinking rates. (a) Histograms of shrinking rates for stationary (red) and moving (blue, v=199 ± 3 nm s^{-1}) microtubules at a kinesin density is 2,500 ± 800 μm^{-2}. (b) A polarity-marked microtubule with a non-fluorescent center segment allows the simultaneous measurement of shrinking rates at the leading and trailing end. (c) Average shrinking rates as a function of the kinesin density (v=199 ± 3 nm s^{-1}) (d) Average shrinking rates as a function of the velocity of microtubules at a kinesin density of 2,500 ± 800 μm^{-2} (circle) and 1,900 ± 600 μm^{-2} (square). The red triangle represents the shrinking rate of stationary microtubules.

The wear process occurs at both ends of the microtubule but not in the center. Measuring the shrinking rate of polarity-marked microtubules (microtubules polymerized with a non-fluorescent center segment, **Figure 26b**)[98] with another batch of kinesins (See further sections) at a density of 2,500 ± 800 μm^{-2} and a gliding velocity of 351 ± 18 nm s^{-1} revealed that microtubules (n=30) shrink from both ends, with the leading end shrinking in average faster than the trailing end (0.88 ± 0.26 nm s^{-1} vs. 0.28 ± 0.10 nm s^{-1}). While the fluorescence intensity in the body region of microtubules decreases by 10-20% during the observation time due to photobleaching and focus drift (see further sections), the rate of decrease is not affected by the presence or absence of gliding (0.74 ± 0.20 $10^{-3} s^{-1}$ at a density of 2,500 ± 800 μm^{-2} and a velocity of 351 ± 18 nm s^{-1} vs. 0.65 ± 0.04 10^{-3} s^{-1} at the same kinesin density and zero velocity).

While events where entire segments with lengths above 150 nm break off the microtubule are not included in the analysis, the length measurements over time do not reveal stepwise changes in length. The distribution of length changes from frame to frame is Gaussian with a standard deviation of 30-40 nm, reflecting the noise in the length measurements. Given the frequency and magnitude of steps implied by our shrinking rates of about 0.5 nm s^{-1}, state-of-the-art algorithms for the detection of steps in such time series can reliably find only step sizes above 30 nm,[99] making it impossible to detect the removal of single or multiple tubulin dimers (See further sections for detailed discussion).

Varying the kinesin density while maintaining the gliding velocity constant at 200 nm s^{-1} reveals a surprising biphasic behavior (**Figure 26c**): At kinesin densities up to 2,000 ± 600 μm^{-2}, the shrinking rate increases only slightly above the shrinking rate of stationary microtubules, while the shrinking rate is five-fold higher at higher kinesin densities. Varying the velocity at kinesin

densities above and below 2,000 ± 600 μm^{-2}, the shrinking rate first increases roughly linearly and then drops after a maximum at 200 nm s^{-1} and 400 nm s^{-1}, respectively (**Figure 26d**).

The pronounced change in the shrinking rate at a density of 2,000 ± 600 μm^{-2} for a velocity of 200 nm s^{-1} (**Figure 26c**) coincides with the experimentally observed mushroom-to-brush transition in the layer of adsorbed kinesins occurring at 1,700 ± 400 μm^{-2}, where the spacing of the kinesins becomes so small that the kinesins begin to constrain and force each other into a stretched brush conformation.[100] This emphasizes that the conformational geometry of the kinesin and the interactions between adjacent kinesins have a significant effect on the wear rate. The shift in the maximal wear rate as a function of the velocity (**Figure 26d**) depends also on the kinesin density, suggesting a complex interdependence between the kinesin density and the microtubule velocity in the wear process. While the initial increase in the wear rate with increasing velocity (**Figure 26d**) is unsurprising, since an increased velocity leads to a proportional increase in microtubule-kinesin contacts per time, the subsequent decrease in wear rates implies that the probability of removing a tubulin dimer decreases rapidly as less time becomes available for each microtubule-kinesin interaction.

A wear process in which individual tubulin dimers are removed from a microtubule (causing undetectable length changes of less than 1 nm) is supported by energetic considerations. Tubulin dimers in the body of the microtubule are stabilized by all available axial and lateral interactions. Tubulin dimers at both ends of the microtubule are missing at least one axial interaction. At least one tubulin dimer at both ends of the microtubule is missing one axial and one lateral interaction, reducing its binding energy by several k_BT and making it exponentially more likely that it will be removed by a force acting on it. At a kinesin density of 3,800 ± 1,200 μm^{-2} and a gliding velocity of 200 nm s^{-1} (where the shrinking rate is maximal) about 20 kinesins per second attach to the leading end and 20 kinesins per second detach from the trailing end while in

average only 0.7 tubulin dimers per second are removed. Thus it is likely that the wear process is focused solely on the tubulin dimers with the weakest attachment.

Kinesin motors exert forces as they interact with the microtubule, both internally when the two motor domains coordinate their actions via mechanical communication,[101] and externally via the attachment to the surface through their tails. Since the kinesin-tubulin interaction of 10-15 k_BT[18] is close in strength to the combined strength of lateral (3-6 k_BT) and longitudinal (7-9 k_BT) tubulin-tubulin interactions[102] of the tubulin dimer having only one lateral and one longitudinal attachment, it is reasonable to observe a non-zero removal probability. The velocity-dependent decrease in detachment rates has been seen previously in competitive unbinding situations and has been identified as a consequence of differences in the shape of the binding energy surface of the competing bonds.[103,104] However, the strongly non-linear dependence of the shrinking rates on kinesin density also demonstrates that the kinesin attachment geometry is critically important. This renders a simple energy barrier crossing model, as employed to explain the atom-by-atom wear observed in sliding studies with atomic force microscopes[17,105-107] insufficient.

In summary, we have shown that the mechanical activity of biomolecular motors can trigger wear at the molecular scale, which in the context of biological nanosystems requires enhanced self-repair mechanisms. Excessive stress or impaired repair may be the cause of medical pathologies originating from wear at the molecular and subcellular level.

Materials and methods

Microtubules were polymerized by reconstituting a 20 µg aliquot of rhodamine-labeled, lyophilized tubulin (TL331M, Lot 357 from Cytoskeleton Inc, Denver, CO) with 6.25 µL polymerization buffer (BRB80 and 4 mM $MgCl_2$, 1 mM GTP, 5% dimethyl sulfoxide), and

incubating it at 37 °C for 30 minutes. The microtubules were then stabilized by diluting them a thousand-fold into BRB80 buffer (80 mM piperazine-N,N'-bis(2-ethanesulfonic acid), 1 mM $MgCl_2$, 1 mM Ethylene Glycol Tetraacetic Acid, pH 6.9 with KOH) with 10 μM paclitaxel (Sigma, St Louis, MO). A kinesin construct consisting of the wild-type, full-length Drosophila melanogaster kinesin heavy chain and a C-terminal His-tag was expressed in Escherichia coli and purified using a Ni-NTA column.[73] Flow cells were constructed using two cover slips and double-sided tape as spacer.[108] For each flow cell, a solution of 0.5 mg mL^{-1} casein in BRB80 buffer was first flown in. After 5 min, the solution was exchanged with the kinesin motor solution (kinesin, 0.5 mg ml^{-1} casein, 1 mM Adenylyl Imidodiphosphate AMP-PNP in BRB80). AMP-PNP (Sigma, St Louis, MO) is an ATP analogue which arrests kinesin action.[47] After another five minutes, this solution was exchanged with microtubule solution containing an enzymatic antifade system[90] (16 nM tubulin, 0.5 mg mL^{-1} casein, 10 μM Paclitaxel, 20 mM D-glucose, 20 μg mL^{-1} glucose oxidase, 8 μg mL^{-1} catalase, 10 mM dithiothreitol, and 1 mM AMP-PNP in BRB80). After another five minutes, the solution of flow cells where microtubules glided was exchanged with a solution containing ATP, an ATP-regenerative system, and an enzymatic antifade system[90] (0.5 mg mL^{-1} casein, 10 μM Paclitaxel, 20 mM D-glucose, 20 μg mL^{-1} glucose oxidase, 8 μg mL^{-1} catalase, 10 mM dithiothreitol in BRB80, 2 mM creatine phosphate, 2 units L^{-1} creatine phosphokinase, ATP, in BRB80). In the case where microtubules were immobilized, the microtubule solution of flow cells was exchanged with AMP-PNP and an enzymatic antifade system (0.5 mg mL^{-1} casein, 100 nM Paclitaxel, 20 mM D-glucose, 20 μg mL^{-1} glucose oxidase, 8 μg mL^{-1} catalase, 10 mM dithiothreitol, 1mM AMP-PNP in BRB80). The openings of the flow cells were then sealed with a small amount of vacuum grease to prevent evaporation of the solutions. All experiments were performed at room temperature.

Microtubules were imaged using a Nikon TE2000-U Epi-fluorescence microscope (Nikon, Melville, NY) equipped with an X-cite 120 lamp (EXFO, Ontario, Canada), an iXON DU885LC EMCCD camera (Andor, South Windsor, CT) and a 100x oil objective (NA 1.45).

For each flow cell, several fields of view were randomly selected and 100 images of each one were taken every 5s with an exposure time of 0.2 s. The FIESTA filament tracking software was then used to measure the length of microtubules over time for each series of images. Microtubule traces were selected to calculate their shrinking rate. The shrinking rates were averaged and the standard error of the mean were calculated and reported on the graphs in **Figure 26**. Each average was calculated from 30 to 100 traces.

To establish that microtubules shrink from both ends, a 20 µg aliquot of HiLyte-labeled microtubules (Cat TL670M, Lot 015 from Cytoskeleton Inc, Denver, CO) was reconstituted with 6.25 µL polymerization buffer, and incubated at 37 °C for 15 minutes. The polymerized HiLyte microtubules were added to a solution where a 20 µg aliquot of rhodamine tubulin (TL331M, Lot 357 from Cytoskeleton Inc, Denver, CO) was suspended in 6.25 µL buffer (BRB80 and 4 mM $MgCl_2$, 1 mM GTP). The solution was then incubated at 37 °C for 8 minutes. The microtubules were then diluted in paclitaxel, imaged, and analyzed as described previously. Kinesins from batch B were used for this experiment (See further sections).

The shrinking rate is measured with a precision below 0.3 nm s^{-1}.

In these experiments, we used kinesin proteins from batch A. We plot the shrinking rates associated with the 40 microtubules tracked by the FIESTA software for a kinesin density of 2,500 ± 785 μm^{-2} and a velocity of 199 ± 3 nm s^{-1} (**Figure 27**). For each microtubule trace, the shrinking rate of the microtubule was calculated by linear regression. The standard deviation of the

Gaussian fit to the histogram of shrinking rates is less than 0.3 nm s^{-1} for both moving microtubules (**Figure 27**) and immobilized microtubules (**Figure 28**).

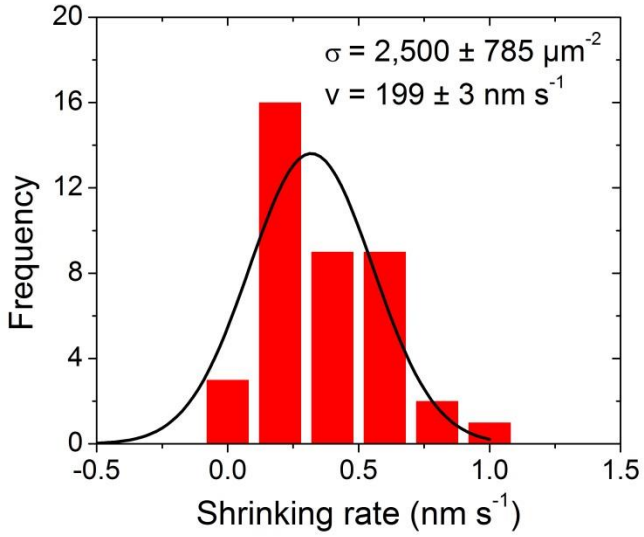

Figure 27. Histogram of 40 shrinking rates calculated from the traces of microtubules using a kinesin density of 2,500 ± 785 μm^{-2} and at a velocity of 199 ± 3 nm s^{-1}. The standard deviation is 0.28 ± 0.05 nm s^{-1} and the center of the peak is at 0.32 ± 0.04 nm s^{-1}.

Figure 28 shows the shrinking rates for the same kinesin density but when microtubules are immobilized on kinesins using AMP-PNP.

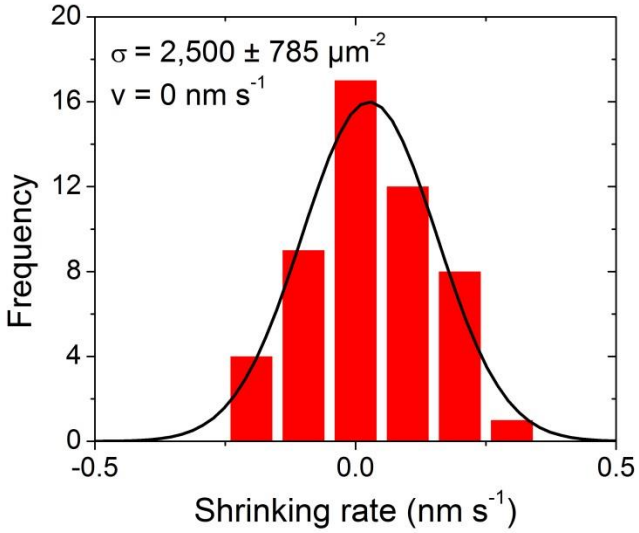

Figure 28. Histogram of 51 shrinking rates calculated from the traces of immobilized microtubules using a kinesin density of 2,500 ± 785 μm^{-2}. The standard deviation is 0.152 ± 0.006 nm s^{-1} and the center of the peak is at 0.027 ± 0.005 nm s^{-1}.

The shrinking rate of immobilized microtubules does not depend on kinesin density.

Using experiments with kinesin batch A, the average shrinking rates of immobilized microtubules was calculated at several kinesin densities (**Figure 29**). When fitting a straight line to the data of **Figure 29**, a Pearson's χ^2 of 0.03467 << 1 is obtained thereby proving that the assumption that the shrinking rate of immobilized microtubules does not depend on kinesin density is consistent with the data of **Figure 29**.

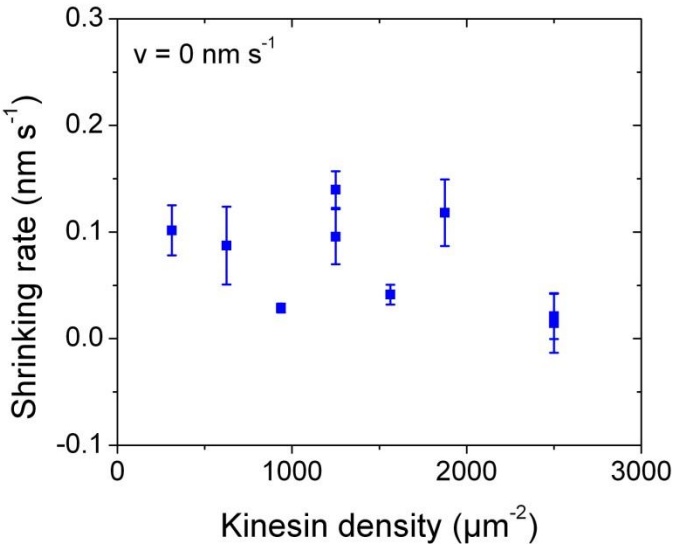

Figure 29. Average shrinking rates at several kinesin densities for immobilized microtubules.

Averaging the 462 data points used to calculate the above average shrinking rates, we find an average shrinking rate of 0.068 ± 0.005 nm s^{-1} for microtubules immobilized on kinesins.

The accuracy of the microtubule length measurement is not affected by movement and there are no discernible "wear" steps.

The frame-to-frame variations in microtubule length for a set of immobilized microtubules (kinesin batch A, kinesin density: 2,500 ± 785 μm^{-2}) are displayed in a histogram in **Figure 30** (1,448 individual steps, 5 nm bins, time between frames 5 s). The frame-to-frame variations in microtubule length follow a Gaussian distribution with a standard deviation of 43 ± 0.8 nm centered at -0.43 ± 0.67 nm (implying an average shrinking rate of 0.09 ± 0.01 nm s^{-1}). The Gaussian distribution reflects the sub-pixel statistical error of the automated length measurement using the FIESTA software package.

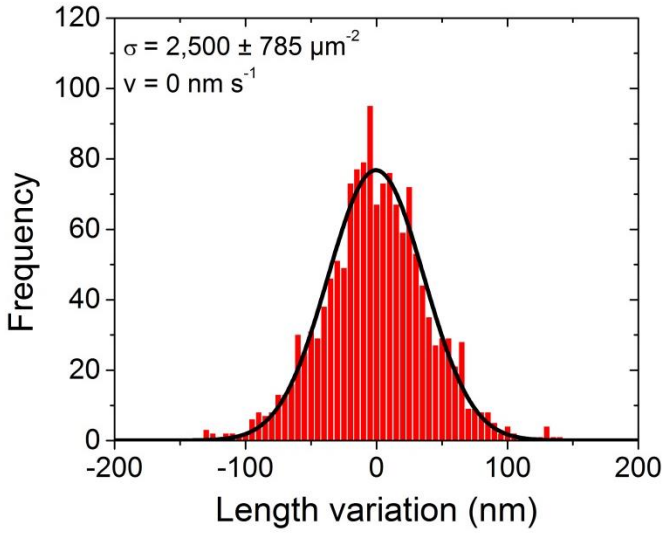

Figure 30. Histogram of 1,448 frame-to-frame length changes (5 s between frames) for a kinesin density 2,500 ± 785 µm^{-2} of and a velocity of 0 nm s^{-1}. The standard deviation is 43 ± 0.8 nm and the center of the peak is at -0.43 ± 0.67 nm.

The frame-to-frame variations in length of moving microtubules (kinesin batch A, kinesin density: 2,500 ± 785 µm^{-2}, velocity: 199 ± 3 nm s^{-1}) are displayed in a histogram in **Figure 31** (1,664 individual steps, 5 nm bins, time between frames 5 s). Again, the frame-to-frame variations in microtubule length follow a Gaussian distribution with a standard deviation of 33 ± 0.5 nm centered at -3.45 ± 0.46 nm (implying an average shrinking rate of 0.7 ± 0.1 nm s^{-1}). Since the movement (and the resulting increased shrinking) did not increase the width of the Gaussian, we conclude that the width again reflects the measurement error.

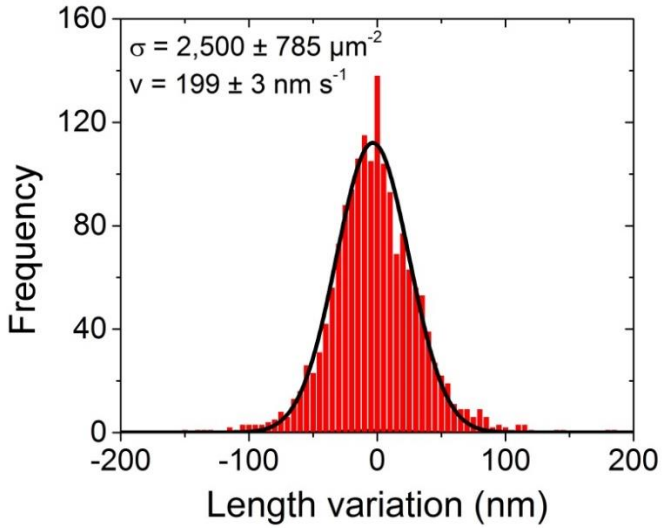

Figure 31. Histogram of 1,664 frame-to-frame length changes (5 s between frames) for a kinesin density 2,500 ± 785 µm^{-2} and a velocity of 199 ± 3 nm s^{-1}. The standard deviation is 33 ± 0.5 nm and the center of the peak is at -3.45 ± 0.46 nm.

The goodness of the fits (Adjusted R^2 are 0.97 and 0.98 for **Figure 30** and **Figure 31** respectively) implies that there are no observable "wear steps". To confirm that, a clustering Bayesian algorithm (described in [100]) was employed which assumed the presence of two Gaussian curves in the histogram with the same standard deviation. However, the algorithm was unable to identify a second Gaussian. To draw the posterior distribution, we used Gibbs sampling, a Markov Chain Monte Carlo algorithm widely used in Bayesian inference.[100]

Breaking events where entire tubular microtubule segments are broken off should also be exceedingly rare: If the probability that a segment is broken off is independent of its length, the observed frequency of breaking events of segments with a length of more than 150 nm (below 10^{-3} s^{-1}) accounts already for more than 90% of all breaking events (given an average microtubule length of more than 2,000 nm) and therefore limits the frequency of breaking off of short segments to less than 10^{-4} s^{-1}. This translates into less than one event in the 1,664 frame-

to-frame length changes fitted here and cannot account for more than 10% of the observed shrinking rate.

Aggarwal et al. discussed the performance of state-of-the-art step detection algorithms and concluded that the number of samples between steps has to be larger than 36 times the square of the ratio of the measurement noise and the step size in order to allow reliable detection of steps.[99] For the present shrinking rate, length measurement noise and frame rate this implies that shrinking events would have to be at least 30 nm in size to be reliably detectable. Since there is no reason to expect steps of this size in our data, the utilization of step detection algorithms would not improve our data analysis. Conversely, to detect the removal of individual tubulin dimers (causing a length change of 8/14 nm), the sampling rate would have to increase to more than 100 kHz, which is infeasible.

At the same time, it is not feasible to visualize the removed tubulin dimer aggregate, because in the 0.3 s it binds to a kinesin,[41] the microtubule would be propelled only 200 nm s^{-1}*0.3 s = 70 nm. Due to the overlap of the microtubule fluorescence signal with the fluorescence from the removed aggregate, it would be impossible to resolve the presence of the aggregate.

In summary, the frame-to-frame length changes are dominated by measurement noise, the probability to remove short tubular segments (breaking events of segments <100 nm) is exceedingly small, and it is practically impossible to detect the individual wear steps.

The fluorescence intensity of microtubules is not impacted by gliding on kinesins.

In these experiments, we used kinesin proteins from Batch B. At a kinesin density of 2,500 ± 825 μm^{-2} and a velocity of 351 ± 18 $nm.s^{-1}$ and 0 $nm.s^{-1}$, we measured the fluorescence intensity of 21 microtubules in their center regions. The microtubule fluorescence intensity per length is

determined by subtracting the integrated counts from a circular region (diameter: 27 pixels) adjacent to a microtubule from the integrated counts of a circular region (diameter: 27 pixels) on a microtubule. We then normalized the fluorescence intensity measured in every frame by the fluorescence intensity measured in the first frame. We show below two typical traces for moving and immobilized microtubules.

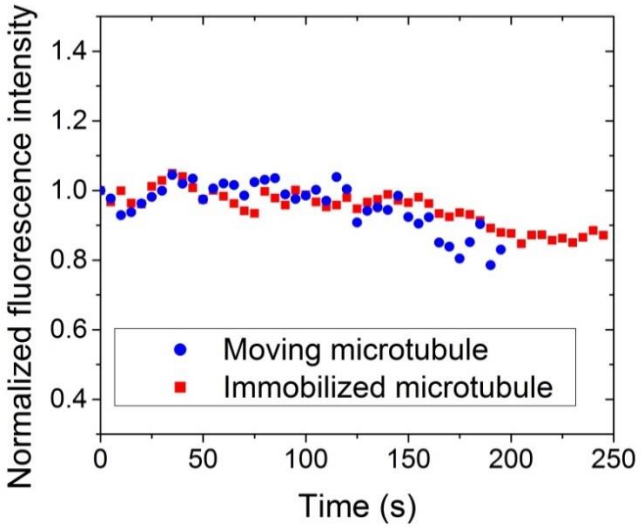

Figure 32. Normalized fluorescence intensity measured in the center of one immobilized microtubule (out of 21) and one moving microtubule (out of 21).

From these 21 traces of moving microtubules and 21 traces of immobilized microtubules, we calculated the rate of decrease in normalized intensity over time for each trace, calculated the weighted average of the 21 rates of decrease in normalized intensity (for both moving and immobilized microtubules), and found that the normalized intensity of both immobilized and moving microtubules decrease at not significantly different rates ($0.65 \pm 0.04\ 10^{-3}\ s^{-1}$ and $0.74 \pm 0.20\ 10^{-3}\ s^{-1}$ respectively), thereby proving that gliding on kinesins does not affect the fluorescence intensity of microtubules.

CHAPTER 4

Wear and breakage combine to mechanically degrade kinesin-powered molecular shuttles

I'm up all night to get lucky

Daft Punk, Random Access Memories

This Chapter will be submitted to Nanoletters as Y. Jeune-Smith, E.L.P. Dumont, and H. Hess, *Wear and breakage combine to mechanically degrade kinesin-powered molecular shuttles*.

Microtubule-based biosensors

In 2009, Fisher et al. showed the proof-of-concept of a "smart dust biosensor" that utilizes microtubules gliding on kinesins,[12] as shown in the figure below.

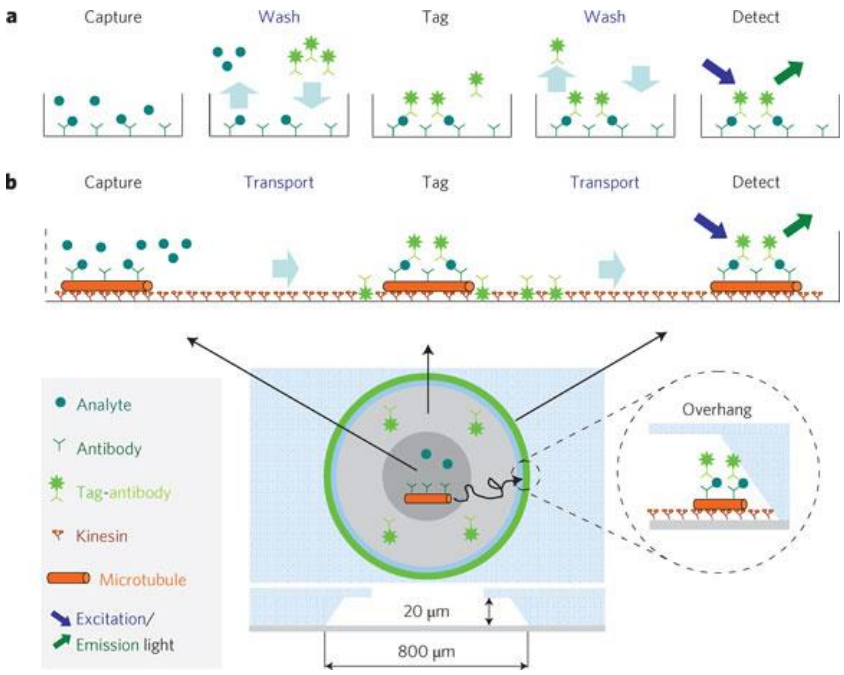

Figure 33. Concept of a smart-dust biosensor utilizing microtubules gliding on kinesins. (a) The capture–wash–tag–wash–detect sequence of a traditional double-antibody sandwich assay. (b) The capture–transport–tag–transport–detect sequence of the smart dust device, in which antibodies on microtubules capture antigens from solution. Kinesin motors are activated, and collisions of antigen-loaded, gliding microtubules with fluorescent particles functionalized with a second antibody lead to pick-up and transfer of the fluorescent tags to the detection zone, indicating the presence of antigen. A basic device layout comprises a circular well created in photoresist on a coverslip. Analyte harvesting, tagging and detection are performed in different radial zones. Adapted from Fisher et al.[12]

If such microtubule-based sensors are to be used for industrial applications, it is important to be able to predict and optimize their lifetime. The wear of microtubules explained in the previous chapter combined with the breaking of microtubules, a phenomenon already known, are enough to predict the lifetime of such microtubule-based biosensors.

Wear and breakage combine to mechanically degrade kinesin-powered molecular shuttles

In macroscopic machines, failure as a result of activation is the result of breakage or wear. Breakage is a sudden and permanent phenomenon often caused by fatigue. Wear, the gradual removal of small amount of material, causes an increasing deviation of the part dimensions from the ideal. Unless breakage intervenes, any system will ultimately fail due to wear. Reducing breakage and wear is a major consideration in machine design.[19]

In molecular and nanoscale machines, breakage and wear will occur as well since the processes have molecular origins.[11] However, state-of-the-art active nanosystems are often operated for only a few cycles to prove their basic feasibility.[109,110] Kinesin-powered molecular shuttles[86,87,111] are hybrid devices, which utilize microtubules and kinesin motor proteins[112] as biological components in a synthetic environment and achieve sustained operation.[113] As a result, they provide an outstanding testbed to observe degradation as a result of activation.

In our shuttle design, kinesin motor proteins are non-specifically adhered to a surface at a density of 2,100±400 μm^{-2} and supplied with their substrate ATP (10-1000 µM) as a source of chemical energy.[114] Microtubules are polymerized from tubulin protein into tubular filaments with an average length of 5 µm and deposited on the kinesin-coated surface. The two “head” domains of the kinesin motors bind to and walk along the microtubules with 8 nm steps, propelling them forward with a velocity between 70 and 500 nm s^{-1} depending on the ATP

concentration.[115] On the order of 250 motors are estimated to attach simultaneously to a microtubule, causing each microtubule to experience up to 30 million motor steps within 5 hours (**Figure 34**).

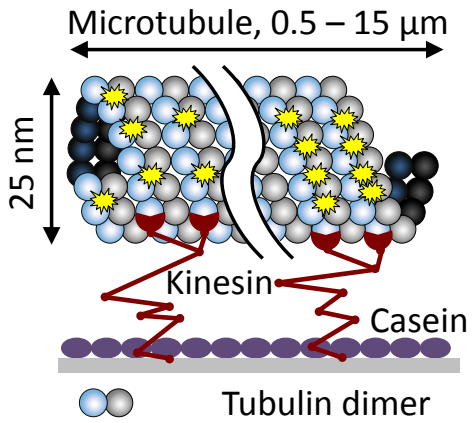

Figure 34. Design of a kinesin-powered shuttle. The yellow stars represent the rhodamine fluorophores.

By observing moving and resting molecular shuttles using fluorescence microscopy, we find that the length and number of microtubules propelled by kinesins on the surface decreases on a time-scale of hours (**Figure 35**, **Error! Reference source not found.**) whereas stationary or free-floating microtubules degrade on a time-scale of days (**Error! Reference source not found.**). Since xidative reactions are known to lead to microtubule degradation, especially during illumination for fluorescence imaging [116], care has been taken to minimize light exposure and control for its effect. To study the effect of active movement, the microtubules are deposited on the surface in the presence of the AMP-PNP, an ATP analogue which arrests motor action [117]. Replacement of the AMP-PNP solution with an ATP solution containing an ATP-regenerating system[118,119] results in a concurrent and constant activation of microtubule gliding. Different ATP concentrations are chosen to achieve gliding velocities at roughly 25, 50, 75 and 100% of the maximum gliding speed

at room temperature. The length of each microtubule in a given area is manually measured for every velocity and time point to determine microtubule length distributions (**Error! Reference ource not found.**). However, the microtubule number per field-of-view and the average length are used as aggregate metrics of the degradation process and plotted as function of time (**Figure 36**).

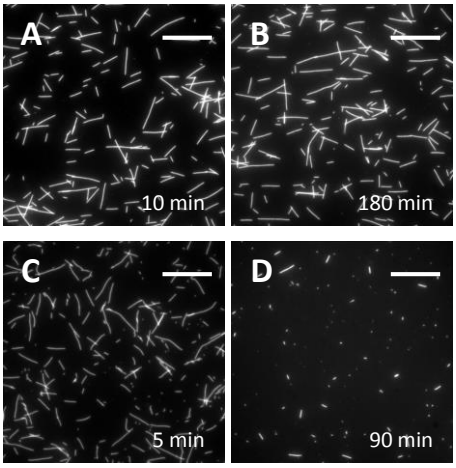

Figure 35. Evidence of degradation (A, B) Rhodamine-labeled microtubules immobilized on surface-adhered kinesin motors by AMP-PNP show no degradation over hours. (C, D) Rhodamine-labeled microtubules gliding on surface-adhered kinesin fueled by 25 µM ATP rapidly degrade in number and length. Scale bar is 20 µm. Experiment by Yoli Jeune-Smith.

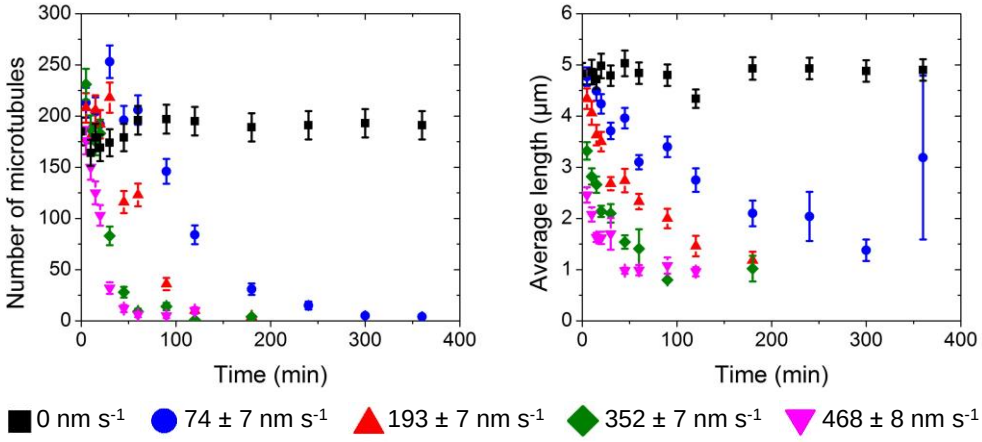

Figure 36. Number and average length of microtubules at several velocities as a function of time. Experiment by Yoli Jeune-Smith.

Examination of microtubule gliding shows that breaking is frequent (**Error! Reference source not ound.**). However, if breaking were the only degradation process, the microtubule number would increase much more significantly in the first ten minutes. Estimates of the microtubule gliding distance before spontaneous detachment due to a trajectory segment without kinesin motors rule out microtubule unbinding as a process responsible for the reduction in microtubule number. Therefore, shrinking is the degradation process responsible for the reduction in microtubule number, because shrinking reduces microtubule lengths to below the 0.5 µm cut-off length for microtubule measurement.

We used a Monte-Carlo simulation which models the degradation of the original distribution of microtubules used for the experiments by both breaking and shrinking the microtubules, and which takes into account the cut-off length (0.5 µm). In the simulation, every microtubule has the same breaking rate and the same shrinking rate per second. Each simulation is run until all microtubules are shorter than the cut-off length, and the results of 500 runs were averaged.

Experimental data for all velocities can be fit with a single breaking rate and a single shrinkage rate, as shown in **Figure 38**. For every velocity, the simulation started from the same population of microtubules. This population of microtubules is composed of 186 microtubules and the average length of the population is 4.79 µm. The initial population of microtubules was calculated by building an average of all histograms of the microtubules populations at zero velocity (width of the histograms is 0.5 µm). The histogram of the initial population of microtubules used in the simulation is shown in **Figure 37**.

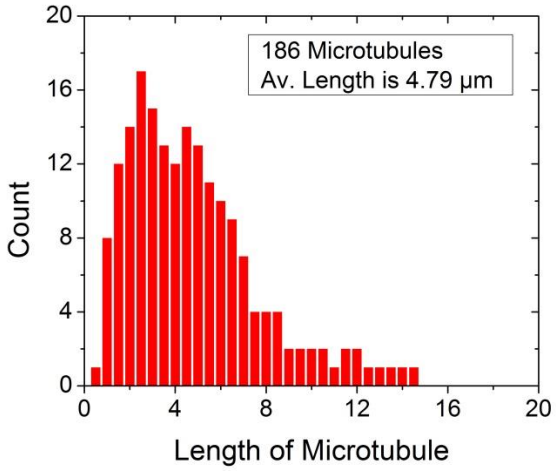

Figure 37. Histogram of the initial populations of microtubules used in the simulations. This population was generated by averaging all populations in the experiment where microtubules were immobilized on the kinesin-coated surface.

In the simulation, each time step represents $\Delta t = 60\text{s}$ in the experiments (but this parameter k can be changed) in order to save computation time. As a result, the shrinking rate in the simulation α_S is related to the shrinking rate in the experiments α_E by:

$$\alpha_S = \Delta t \times \alpha_E$$

Since $\Delta t = 60\text{s}$, the breaking rate in the simulation β_S is equal to the breaking rate per minute in the experiments β_E by:

$$\beta_S = \beta_E$$

For each velocity, I chose the combination of shrinking rate and breaking probability that minimizes the average of the 1,000 sums of the errors corresponding to the curve of the number of microtubules and the curve of their average length. This error is defined for each velocity v by the following formula:

$$\varepsilon = \sum_{i=1}^{n_v} \left(\frac{N_{i,S} - N_{i,E}}{N_{i,E}} \right)^2 + \sum_{i=1}^{n_v} \left(\frac{A_{i,S} - A_{i,E}}{A_{i,E}} \right)^2$$

Where N_S and N_E are respectively the simulated and experimental arrays of the number of microtubules for a given velocity. The number of data points in these arrays is n_v. A_S and A_E are respectively the simulated and experimental arrays of the average of microtubules for a given velocity v.

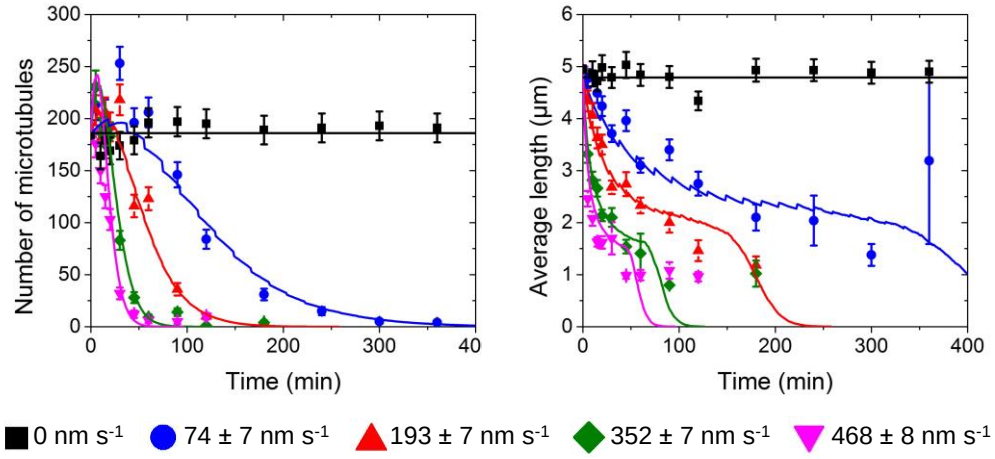

Figure 38. Monte Carlo simulations on the number and average length of microtubules at several velocities as a function of time. Experiments by Yoli Jeune-Smith.

We then plotted the shrinking rate and the breaking probabilities as a function of the velocity, as shown in the figure below.

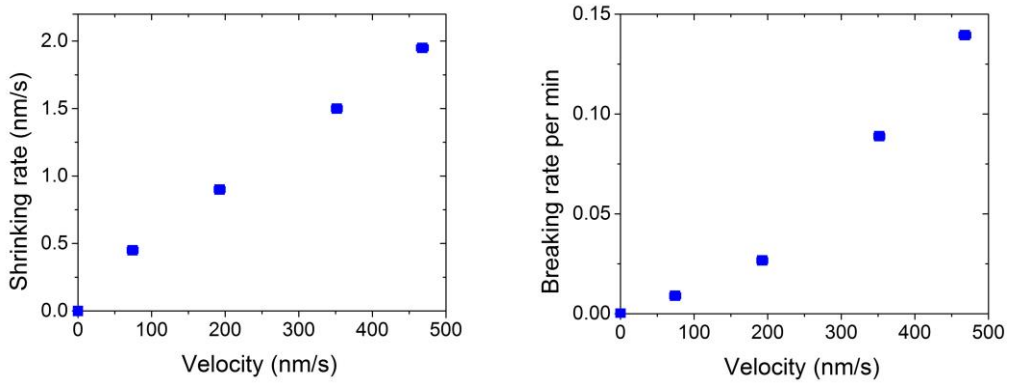

Figure 39. Shrinking rate and breaking probability found by the Monte Carlo simulations as a function of velocities.

The frequency of breaking matches the expected frequency of encountering a defective kinesin motor (~1 out of 2,000 [96]). These encounters result in temporary binding along the microtubule, build-up of strain as the functional motors attached to the microtubule continue to propel the microtubule, and eventually breaking (**Error! Reference source not found.**). We have not found vidence that fatigue causes breakage despite the high number of motor steps that a microtubule experiences. Instead, the defective motors act almost like "potholes" on the microtubule path, and a method to remove defective motors would reduce the frequency of binding events.

Subcellular systems, such as the cytoskeleton, are prototypical active nanosystems and motor-induced disassembly of the cytoskeleton has recently been observed.[120] Evolution must have developed successful strategies to minimize and also to counteract molecular wear, for example by constant replacement of supramolecular structures.

Finally, probing the relation between force, lifetime and chemistry in single molecular bonds[121] has been an extremely fruitful pursuit in nanoscience. The emerging challenge is to transfer our insights to assemblies of molecules experiencing complex force spectra, in order to build a molecular theory of tribology.

Materials and Methods

The experiments were done by Yoli Jeune-Smith and are explained in her PhD thesis.

MATLAB code to predict the degradation of microtubules gliding on kinesins.

The structure of the MATLAB code is given below.

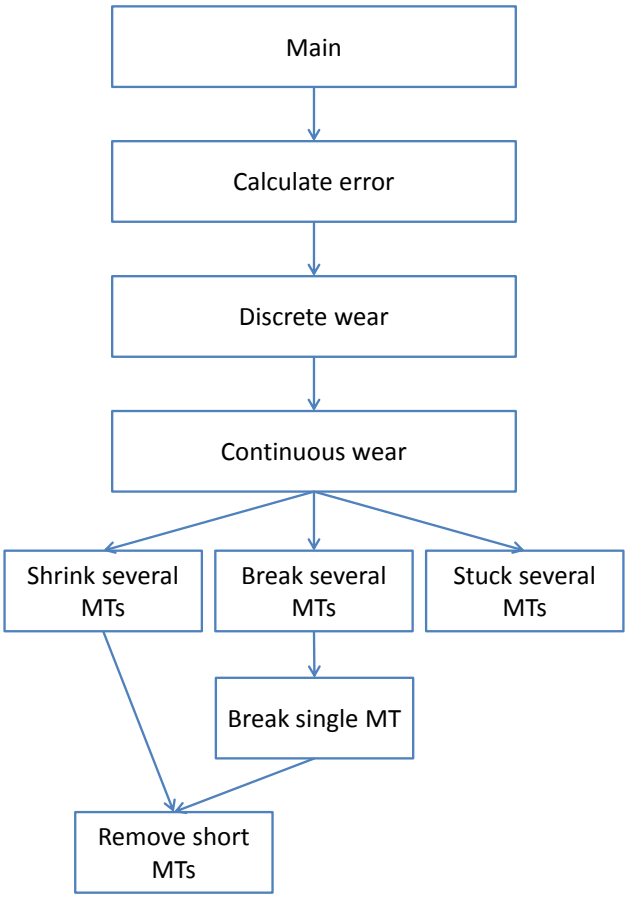

Figure 40. Structure of the MATLAB code to predict the wear of microtubules gliding on kinesins as a function of a shrinking rate in nm s^{-1} and a breaking probability per second.

Each "box" in **Figure 40** represents a different file. To launch the program, the user has to launch the file "main". In this file, the user choses the shrinking rates and breaking probabilities to be used in the simulations. The file "Calculate Error" calculates the error of the simulated wear compared to the experimental data points for all velocities. These errors are stored in the MAT file "simulation.mat". "Continuous wear" generates the average number and the average length of the initial set of microtubules every time interval that was chosen in the file "main" (variable 'time_step'). "Discrete wear" extracts from the wear generated by "Continuous wear" the average length and the average number of microtubules corresponding to the different time

points at which measurements were made during the experiments. The modules “Shrink”, “Break”, “Remove”, and “Stuck” are self-explanatory: they respectively shrink, break, remove and stop microtubules in the given population.

“main.m”

```
%% CHOSE EXPERIMENTAL DATA

clear all;
close all;

load('simulation.mat');
eval( strcat('simu = simulation'));

%%

% In the simu.parameters file
% Column 1 = Shrinking rate (nm/s)
% Column 2 = Breaking probability (per min)
% Column 3 = Length cut off for a MT under which MTs are ignored in
% simulation
% Column 4 = Time step in simulation (>1s)
% Column 5 = Number of loops over which simulations are run
% Column 6 = Error of the simulation when compared to 10 uM data points.
% Column 7 = Error of the simulation when compared to 25 uM data points.
% Column 8 = Error of the simulation when compared to 80 uM data points.
% Column 9 = Error of the simulation when compared to 1000 uM data points.


% In the simu.continuous file:
% Column 1: time in min
% Column 2: nb of MTs
% Column 3: av length of MTs

%% SIMULATION parameters

global                    nb_of_loops          ; % number of loops in the MC simulation
global                    max_time             ; % maximum time over which the simulation is ran
global                    stuck_cut_off        ; %Cut-off length for a mt to have a chance to get stuck
global                    time_step            ;
global                    record_continuous_wear ;
global                    plot_wear_graphs     ;
global                    length_cut_off       ;
global                    generate_wear_anyway ;
global                    combination_already_exists ;
global                    continuous_wear_to_record ;

nb_of_loops               =    500              ;

max_time                  =    400*60           ; % in seconds

stuck_cut_off             =    1                ; % in um

time_step                 =    60               ; % in s. Must be > 1.

record_continuous_wear    =    true            ; % requires a lot of space

plot_wear_graphs          =    false            ;

generate_wear_anyway      =    false            ; % To test reproducibility of the runs

length_cut_off            =    0.5              ; % Minimal size of MTs to be considered (in um)


%% DEFINITION OF THE SHRINKING RATE AND BREAKING RATE

global                    shrinking_rate       ; % shrinking rate of MTs in nm/s
global                    breaking_proba       ; % proba that MTs break per s
global                    stuck_proba          ; % proba that MTs get stuck per s

stuck_proba               =    0               ;

nb_iterations_shrinking   =    0               ; % 0 if only one shinrking rate is wanted
nb_iterations_breaking    =    0               ; % 0 is only one breaking rate is wanted

lowest_shrinking_rate     =    1.95               ;  % nm/s
highest_shrinking_rate    =    1.6                ;  % nm/s

lowest_breaking_proba     =    0.139453661        ;  % probability per minute (time_step need to be 60s)
highest_breaking_proba    =    0.18               ;  % probability per minute (time_step need to be 60s)
```

```
step_shrinking            =    (highest_shrinking_rate - ...
                               lowest_shrinking_rate)/nb_iterations_shrinking;

step_breaking             =    (highest_breaking_proba - ...
                               lowest_breaking_proba)/nb_iterations_breaking;

shrinking_rate            =    lowest_shrinking_rate ;
breaking_proba            =    lowest_breaking_proba ;

%% LOOP OVER ALL COMBINATIONS OF SHRINKING RATE AND BREAKING RATE

best_shrinking_rate       =    lowest_shrinking_rate ;
best_breaking_proba       =    lowest_breaking_proba ;
best_error                =    1e12             ;

for i=0:nb_iterations_shrinking
    for j=0:nb_iterations_breaking

        disp(['Shrinking ',num2str(i),'/',num2str(nb_iterations_shrinking)]);
        disp(['Breaking ', num2str(j),'/',num2str(nb_iterations_breaking)]);
        disp(['Shrinking rate is ',num2str(shrinking_rate)]);
        disp(['Breaking rate is ',num2str(breaking_proba)]);
        disp('--------------');

        % Find if the combination of simu.parameters already exists
        combination_already_exists = false ;
        indice_if_already_exists   = 1     ;

        for k=1:simu.indice-1
          if simu.parameters(k,1) == shrinking_rate && ...
             simu.parameters(k,2) == breaking_proba && ...
             simu.parameters(k,3) == length_cut_off && ...
             simu.parameters(k,4) == time_step      && ...
             simu.parameters(k,5) == nb_of_loops
                combination_already_exists   =  true ;
                indice_if_already_exists     =  k    ;
          end
        end


        if combination_already_exists == false || generate_wear_anyway == true


            sse = Calculate_Error() ;

            % Writes the simu.parameters and the error in the spreadsheet:
            simu.parameters(simu.indice,:) = [shrinking_rate, breaking_proba,...
                                              length_cut_off,time_step,...
                                              nb_of_loops,sse] ;

            % Save the simulations in case the algorithm crashes
            if record_continuous_wear == true
                eval( strcat('simu.continuous.s',num2str(simu.indice),...
                             '=continuous_wear_to_record')) ;
            end

            % Increment the row available for simu.parameters:
            simu.indice    =   simu.indice + 1 ;

        end

        breaking_proba = breaking_proba + step_breaking;

        % Save simulation
        eval( strcat('simulation','= simu'));
        eval( strcat ('save simulation.mat simulation'));


    end

    breaking_proba = lowest_breaking_proba ;
    shrinking_rate = shrinking_rate + step_shrinking;
end

clear all;
```

"Calculate_Error.m"

```
function [sse] = Calculate_Error()

load('yoli_data.mat');
```

```
global                    generate_wear_anyway ;
global                    combination_already_exists ;
global                    time_array           ; % in s
global                    nb_MTs               ;
global                    error_nb_MTs         ;
global                    av_length            ; % in um
global                    error_av_length      ;

sse                   =   zeros(1,4)           ;

[nb_MTs_continuous,av_length_continuous] = Continuous_wear();

for k=1:4
    if k==1
        data_name     =   'um10'      ;
    elseif k==2
        data_name     =   'um25'      ;
    elseif k==3
        data_name     =   'um80'      ;
    elseif k==4
        data_name     =   'um1000'    ;
    end

% Store the wanted data set in the array 'data'
eval( strcat('data = yoli_data.',data_name,';'));

% Extracts the different columns from the data
time_array                =    data(:,1)       ;
nb_MTs                    =    data(:,2)       ;
error_nb_MTs              =    data(:,3)       ;
av_length                 =    data(:,4)       ;
error_av_length           =    data(:,5)       ;

% Get the simulated wear at the specific times of the wear data points
if combination_already_exists == false || generate_wear_anyway == true
            [nb_MTs_simu,av_length_simu] = Discrete_wear(nb_MTs_continuous,av_length_continuous);

            % Calculate the error on the number of MTs and their av. length
            nb_MTs_error      =    sum(((nb_MTs_simu - nb_MTs)./nb_MTs).^2)    ;
            av_length_error   =    sum(((av_length_simu - av_length)./av_length).^2) ;
            sse(k)               =    nb_MTs_error + av_length_error ;


end

end
```

“Discrete_wear.m”

```
function [nb_MTs_discrete,av_length_discrete] = Discrete_wear(nb_MTs_continuous,av_length_continuous)

global                    time_array           ;
global                    time_step            ;

%%


nb_MTs_discrete           =    zeros(length(time_array),1) ;
av_length_discrete        =    zeros(length(time_array),1) ;

nb_MTs_discrete(1)        =    nb_MTs_continuous(1)     ;
av_length_discrete(1)     =    av_length_continuous(1)  ;

position_in_array         =    2                        ;
current_time              =    time_step                ;

for i=2:length(time_array) % i=1 was treated above (t=0)

    while abs(current_time - time_array(i)) > time_step/2
       current_time       =    current_time + time_step ;
       position_in_array  =    position_in_array + 1    ;
    end
    if position_in_array <= length(nb_MTs_continuous)
       nb_MTs_discrete(i) = nb_MTs_continuous(position_in_array) ;
       av_length_discrete(i)  =    av_length_continuous(position_in_array);
    end

end

end
```

"Continuous_wear.m"

```
function [nb_MTs_simu_looped,av_length_simu_looped] = Continuous_wear()
% Store the values of number of array_of_mts in an excel file

load('initial_distribution.mat');


global                    max_time                          ;
global                    time_step                         ;
global                    time_array                        ;
global                    shrinking_rate                    ;
global                    breaking_proba                    ;
global                    stuck_proba                       ;
global                    nb_of_loops                       ;
global                    nb_MTs                            ;
global                    av_length                         ;
global                    error_nb_MTs                      ;
global                    error_av_length                   ;
global                    record_continuous_wear            ;
global                    plot_wear_graphs                  ;
global                    continuous_wear_to_record         ;


max_size_array            =    round(max_time/time_step)     ;
array_of_loops            =    zeros(max_size_array,1)       ;

nb_MTs_simu_looped        =    zeros(max_size_array,1)       ;
av_length_simu_looped     =    zeros(max_size_array,1)       ;

shrinking_rate_simulation =    1e-3*shrinking_rate*time_step  ;

breaking_proba_simulation =    breaking_proba                ;

stuck_proba_simulation    =    time_step*stuck_proba/(...
                               time_step*stuck_proba+1-stuck_proba);

%% Simulate the continuous wear over nb_of_loops times


for i=1:nb_of_loops

  %disp('------------------------------');
  disp(['loop ',num2str(i)]);

  % Definition of variables
  nb_MTs_simu        =   zeros(max_size_array,1)  ;
  av_length_simu     =   zeros(max_size_array,1)  ;

  % initial conditions
  array_of_mts       =   initial_distribution     ;
  nb_MTs_simu(1)     =   length(array_of_mts)     ;
  av_length_simu(1)  =   mean(array_of_mts)       ;
  current_time       =   time_step                ;
  indice_in_array    =   2                        ;

  while isempty(array_of_mts)==0 && current_time<max_time

    %disp(['Current time is ',num2str(current_time)]);

    % Shrink MTs
    array_of_mts    =   Shrink_several_MTs(array_of_mts,shrinking_rate_simulation);
    %disp('MTs were shrunk');

    % Stuck MTs
    array_of_mts    =   Stuck_several_MTs(array_of_mts,stuck_proba_simulation);
    %disp('MTs were stuck');

    % Break MTs
    array_of_mts    =   Break_several_MTs(array_of_mts,breaking_proba_simulation);
    %disp('MTs were broken');

    if isempty(array_of_mts) == 0 % if there are still MTs in the array
    nb_MTs_simu(indice_in_array)     =   length(array_of_mts)      ;
    av_length_simu(indice_in_array)  =   mean(array_of_mts)        ;
    end
    current_time    =   current_time + time_step                   ;
    indice_in_array =   indice_in_array + 1                        ;
  end

  %disp('Got out of the while loop');

  % Add the simulated wear in the final array that will be averaged later.
  for j=1:min(length(nb_MTs_simu),max_size_array)
      nb_MTs_simu_looped(j) = nb_MTs_simu_looped(j)+nb_MTs_simu(j);
      av_length_simu_looped(j) = av_length_simu_looped(j)+av_length_simu(j);
      array_of_loops(j) = array_of_loops(j)+1;
  end
```

```
  %disp('stored the fresh data');

end

% Average all the loops
for i=1:max_size_array
    nb_MTs_simu_looped(i) = nb_MTs_simu_looped(i)/array_of_loops(i);
    av_length_simu_looped(i) = av_length_simu_looped(i)/array_of_loops(i);
end

% Delete the zeros and the NaN
nb_MTs_simu_looped(nb_MTs_simu_looped==0) = [] ;
av_length_simu_looped(av_length_simu_looped==0) = [] ;
nb_MTs_simu_looped(isnan(nb_MTs_simu_looped)) = [];
av_length_simu_looped(isnan(av_length_simu_looped)) = [];


timeframe =  0:time_step:length(nb_MTs_simu_looped)*time_step-1;
timeframe = timeframe';

%% Record the simulated wear

if record_continuous_wear == true
    timeframe_in_min = timeframe./60;
    continuous_wear_to_record = [timeframe_in_min,nb_MTs_simu_looped,av_length_simu_looped];
end



%% Plot graphs

if plot_wear_graphs == true

    linewidth                =  2        ;

    subplot(121)
    plot(timeframe,nb_MTs_simu_looped,'color','red','linewidth',linewidth);
    hold on;
    errorbar(time_array,nb_MTs,error_nb_MTs,'x');
    axis([0 1.1*max(time_array) 0 300]);
    xlabel('Time (s)','fontsize',14);
    ylabel('Number of MTs','fontsize',14);
    title('NUMBER OF MICROTUBULES','fontweight','b','fontsize',16)

    subplot(122)
    plot(timeframe,av_length_simu_looped,'color','red','linewidth',linewidth);
    hold on;
    errorbar(time_array,av_length,error_av_length,'x');
    axis([0 1.1*max(time_array) 0 6]);
    xlabel('Time (s)','fontsize',14);
    ylabel('Average Length of MTs in um','fontsize',14);
    title('AVERAGE LENGTH','fontsize',16,'fontweight','b')

end



end
```

"Shrink_several_MTs.m"

```
function [microtubules] = Shrink_several_MTs(mts,amount_to_be_shrunk)

  microtubules = mts - amount_to_be_shrunk ;

  microtubules = Remove_short_mts(microtubules) ;

end
```

"Break_several_MTs.m"

```
function [mts_broken] = Break_several_MTs(mts,proba_break)
  % Breaks an array of MTS

mts_broken = [];

for i=1:length(mts)
    mts_broken = [mts_broken;...
                  Break_single_mt(mts(i),proba_break)];
end
```

```
end
```

“Break_single_MT.m”

```
function [final_lengths] = Break_single_mt(length_MT, proba_break)
% Breaks one MT

  random_number = rand();

 if(random_number<=proba_break)
     random_number_mt =  rand()                          ;
     broken_piece_1   =  random_number_mt*length_MT      ;
     broken_piece_2   =  length_MT-broken_piece_1        ;
     final_lengths    =  Remove_short_mts([broken_piece_1,broken_piece_2]);
 else
     final_lengths    =  [length_MT]                     ;
 end;
```

“Stuck_several_MTs.m”

```
function [mts_not_stuck] = Stuck_several_MTs(mts,stuck_probability_simulation)

global stuck_cut_off ;

mts_not_stuck = [];

for i=1:length(mts)
    if mts(i)<=stuck_cut_off
         random_probability = rand();
         if random_probability > stuck_probability_simulation
            mts_not_stuck = [mts_not_stuck;mts(i)];
         end
    else
         mts_not_stuck = [mts_not_stuck;mts(i)];
    end
end


end
```

“Remove_short_mts.m”

```
function microtubules = Remove_short_mts(mts)
  %Clean array of MTs if MTs have a length inferior to 0.5 um

  global length_cut_off                          ;
  indice            =   1                         ;
  microtubules      =   zeros(length(mts),1)      ;

  for i=1:length(mts)
    if mts(i)>=length_cut_off
        microtubules(indice) = mts(i);
        indice = indice+1;
    end;
  end


  microtubules(microtubules==0) = [] ;
```

CHAPTER 5

Fatigue failure and molecular machine design

The world is a dynamic mess of jiggling things, if you look at it right

Richard Feynman

This Chapter was published as H. Hess and E.L.P. Dumont, *Fatigue Failure and Molecular Machine Design*, Small, **2011**, 7, 1619-1623.

Introduction

In nature, molecular machines have evolved to perform a wide range of tasks within the cellular factory, including assembly, transport, and actuation.[122] Advances in single molecule studies over the past two decades have enabled the deduction of the mechanisms of operation, revealing for example the head-over-head movement of kinesin motor proteins,[123] the stepwise growth of microtubules,[124] and the reversible movement of RNA-processing machinery[125]. Careful attention has been paid to the coupling of mechanical and chemical cycles .[126] Applying these insights, synthetic molecular machines, including rotaxanes[127] and DNA motors[128], have been designed to combine chemical and mechanical cycles to fulfill functions such as actuation.[129]

However, machine design in engineering is a process in which the mechanism of operation is only the starting point.[19] In the design process, the static and dynamic loads on all machine parts are determined, and the dimensions and materials of these parts are specified to sustain these loads within the framework of theories of static failure, surface failure, and fatigue failure.

A car engine is an excellent illustration of the need to consider not only static failure ("Will it break when turned on the first time?") but of surface failure ("When will the piston wear out?") and fatigue failure ("When will the crankshaft break?"). An unused engine will of course eventually fail due to oxidation ("rusting"), but materials selection and external measures, such as encapsulation, have reduced oxidation to an acceptable level.

The challenge of fatigue failure at the molecular scale

At the molecular scale, the presence of thermal forces complicates the assignment of a specific load to a molecular structure. In atomic force microscopy experiments, where single intermolecular bonds were subjected to loads increasing at varying rates,[130] the rupture force was found to be dependent on the rate of loading. The relationship between loading rates and

rupture forces was explained using unimolecular reaction rate theory, which accounts for the thermal forces assisting the breaking of the bond.[121] In brief, the external load deforms the potential energy landscape of the bond, and thereby lowers the activation energy for unbinding, accelerating the unbinding rate according to the Arrhenius equation. Interestingly, the effect of thermally driven breaking of bonds under stress has been observed at the macroscale more than forty years earlier, when Zhurkov measured the lifetime of macroscopic samples under load.[131]

While Evans' conceptual framework[121] has been successfully applied to predict the failure of intra- and intermolecular bonds under static or linearly increasing loads, molecular machines operate over many cycles of time-varying loads. At the macroscale, fatigue failure can occur at a stress of little more than half of the ultimate stress when the part is subjected to stress over millions of cycles.[19] The development of theories to predict fatigue failure has been a major advance in 20th century engineering and has enabled a substantial refinement of macroscale machinery.

To appreciate the number of cycles experienced by a molecular machine such as a myosin motor protein, consider a human heart muscle, which contracts more than 2 billion times over its 80-year lifespan. This roughly equals the number of rotations of a modern piston engine over its lifetime (**Figure 41**).

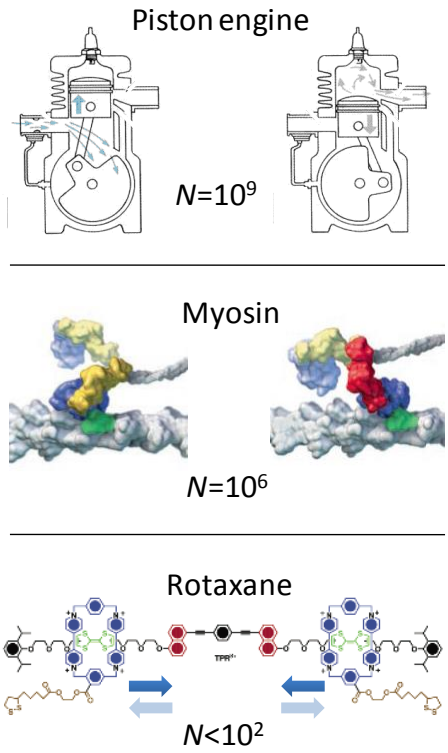

Figure 41. Mechanical fatigue at different scales. While piston engines achieve lifetimes exceeding a billion revolutions, myosin II motor proteins are replaced after a million cycles[132] (Figure adapted from [133]), and arrays of rotaxanes experience diminished force output after a few dozen cycles[134] (Figure adapted from [134]).

However, individual myosin motor molecules are replaced within days,[132] meaning they operate "only" for about a million cycles. Artificial muscles assembled from synthetic molecular motors lose their contractility after a few dozen cycles[134]. This raises questions: Which design constraints drove evolution towards molecular motors operating for millions of cycles while being replaced a thousand times rather than designing molecular motors capable of operating for billions of cycles in the first place, and what design changes have to be made to synthetic molecular motors to achieve sustained operation over large numbers of cycles? In essence, the mastery of machine

design at the molecular scale is a precondition for successful engineering applications of molecular machines, such as molecular motors.

Macroscopic fatigue failure theories model parts as being continuous solids with pre-existing microscopic cracks where crack propagation is initiated, leading ultimately to mechanical failure (**Figure 42**). In low cycle fatigue design (LCF, N<1,000), fracture mechanics is employed to predict and limit crack propagation. The objective in high cycle fatigue design (HCF, N>1,000) is typically to lower the stress below the endurance limit of the material. The stochastic nature of failure is captured by reliability factors which reduce the fatigue strength (e.g. for steel: C_{reliab}(99.999%) = 0.659).[19]. These concepts are difficult to apply at the molecular scale, where the parts cannot be approximated as continuous solids and where thermal activation plays a major role.

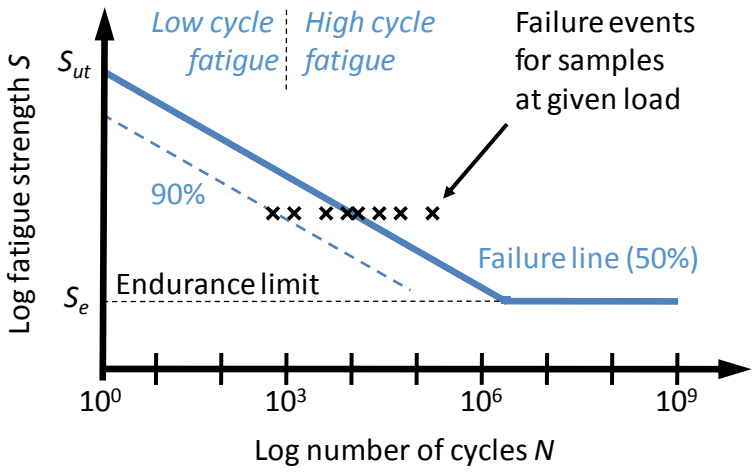

Figure 42. S-N diagrams are generated by exposing parts to cycles of defined load until failure. The number of cycles varies from part to part, and the failure line indicates the number of cycles after which 50% of the samples have failed for a given load. Some materials (e.g. steel) exhibit an endurance limit S_e below which fatigue failure is not observed. Other materials experience fatigue failure eventually under any load. To increase the reliability of the part (e.g. to 90%), the stress has to be reduced by a reliability factor C_{reliab} relative to the fatigue strength $S(N)$.

Mechanical fatigue of a molecular bond

Approaching fatigue from the other extreme, the response of an individual bond to cyclic stress can be evaluated. Over the course of a force application cycle, the unbinding rate varies according to:

$$k(t) = k_0 \times \exp\left(-\frac{U_0 - F(t)x^*(t)}{k_B T}\right) \qquad (1)$$

This generalized Kramers relation assumes that the frequency of force application is much smaller than the bond frequency k_0, which usually applies.[121,135]

For a bond subjected to a force alternating between two constant values of tension and compression (See **Figure 42**), the compression cycle cannot unbind, and the average rate becomes simply:

$$\langle k \rangle = \frac{k_0}{2} \times \exp\left(-\frac{U_0 - Fx^*}{k_B T}\right) \qquad (2)$$

Figure 43 illustrates this for a bond described by a Lennard-Jones Potential with $U_0 = 40\ k_B T$ and r_m = 0.5 nm. The time at which in average half of the bonds still survive is given by $t = \ln(2)/\langle k \rangle$.

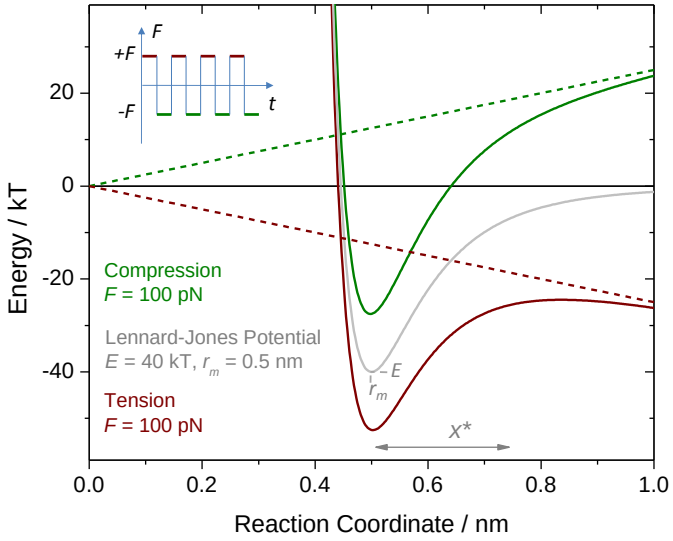

Figure 43. Cyclic tension and compression will deform the potential energy surface of a bond and facilitate the rupture of the bond during the tension phase of the cycle.

Therefore, each 10-fold increase in average lifetime requires a reduction of 2.3 $k_B T/x^*$ in the applied force **(Figure 44)**. The number of force cycles is not a factor here, in contrast to macroscopic systems where the number and magnitude of stress cycles rather than the time-integrated stress is seen as the primary factor in determining failure.

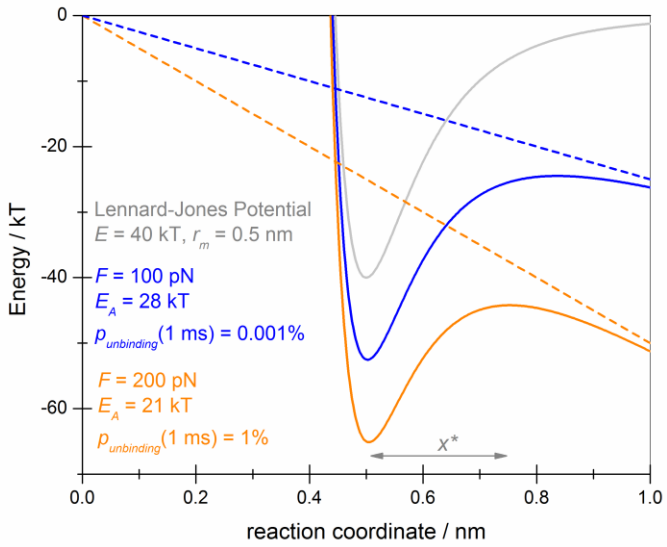

Figure 44. The deformation of a hypothetical Lennard-Jones potential (*E* = 40 kT, r_m = 0.5 nm) by a constant force. As the force is increased from 100 pN to 200 pN, according to eq. (1) the probability for the hypothetical bond to remain intact after 1 ms falls from 99.999% to 99%.

Similar to the use of reliability factors in macroscale fatigue calculations, the applied force has to be reduced to ensure that a larger fraction of bonds reaches the specified lifetime. If a bond is designed to reach a certain lifetime for an applied force of F_{90} with 90% probability, the force has to be reduced by 2.3 k_BT/x^* for a ten-fold increase in reliability (99%). Given that forced unfolding single molecule experiments have revealed distances to the transition state x^* of as little as 0.1 nm in proteins,[136] a ten-fold increase in reliability implies a reduction in applied force of as much as 90 pN, which is not insignificant relative to unfolding forces of 300-500 pN observed in AFM experiments with loading rates of hundreds of nm s^{-1}.

In the case of myosin, which executes a single powerstroke in 1 ms, eq. (2) suggests that the barrier to unfolding needs to be only about 17 k_BT to survive a single powerstroke (k_0

assumed to be 10^{-10} s^{-1}). However to achieve operation for a million cycles, the barrier to unfolding in the presence of load has to be close to 30 k_BT. This is a significant challenge for the molecular design since a hydrogen bond has a bond energy of only 10 k_BT.[137]

An interesting corollary of eq. (1) is that the product of a constant sustained force and the force-dependent lifetime is maximized for an applied force equal to k_BT/x^*, independent of the bond strength. At this force, the lifetime is reduced only by a factor of e relative to the lifetime of the unloaded bond, which can make it difficult to discern the reduction in lifetime due to the applied force. As a result, the lifetime of a bond or an entire molecular machine may appear to be limited by thermally-driven unbinding or disintegration, when in fact the design is optimized to maximize sustained force and lifetime simultaneously.

Increasing the lifetime of molecular bonds

The longevity of molecular structures may be significantly extended by rebinding, in contrast to macroscopic cracks, which do not close under compression. Without rebinding, two parallel bonds break sequentially providing the structure a lifetime of

$$\langle t \rangle = \frac{1}{k_1} + \frac{1}{k_2} \approx \frac{1}{k_1} \qquad (3)$$

where k_1 and k_2 are the unbinding rates for the first and second bond, respectively. Since the second bond experiences the full load after the first bond broke, k_2 >> k_1. With rebinding, the first broken bond can return to a bound state, prior to the unbinding of the second bond, providing a lifetime of

$$\langle t \rangle = \frac{k_1 + k_{-1} + k_2}{k_1 k_2} \approx \frac{1}{k_1}\left(1 + \frac{k_{-1}}{k_2}\right) \qquad (4)$$

Since there is no barrier to rebinding (**Figure 42**) and since cyclic loading brings the binding partners in close vicinity during the compression phase, the rebinding rate k_{-1} will exceed the unbinding rates of the first bond k_1 (experiencing a force *F*/2) and the second bond k_2 (experiencing the full force *F*) by orders of magnitude, providing a highly effective mechanism to extend the lifetime **(Figure 45)**.

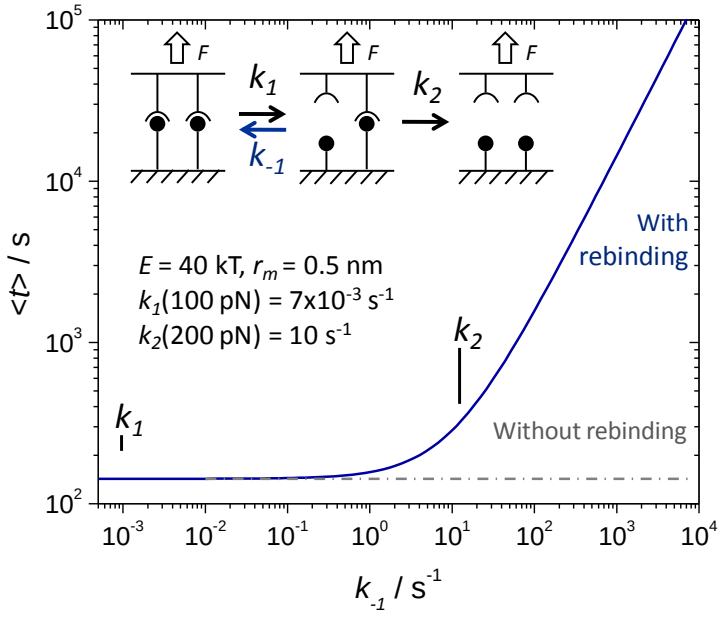

Figure 45. The effect of rebinding on the average time to break two parallel bonds. The lifetime of the double-bond increases rapidly as the rebinding rate exceeds the unbinding rate of the second bond under full load. Since rebinding is facilitated by compression or at least absence of tension, the frequency of the force cycle becomes an important parameter.

This line of reasoning can be applied to polyvalent interactions in general.[138] Bonds with multiple energy minima along their reaction coordinate, such as the biotin-streptavidin bond, can similarly transition back to their most stable configuration after partial unbinding.[103,139] Due to facilitated rebinding, cyclic loading of parallel bonds should lead to drastically different behavior from static

loading of equal duration. Of course, the general increase in lifetime is also obtained via other mechanisms to reverse partial failure, such as refolding of a partially unfolded protein.

A second mechanism to extend the lifetime of supramolecular structures is continuous turnover. For cytoskeletal filaments, a dynamic equilibrium between assembly and disassembly enables complex functions, such as the search for, capture and transport of chromosomes by microtubules during mitosis.[140] At the same time, turnover may eliminate growing defects and maintain these nanoscale mechanical structures in pristine condition. "Treadmilling" of microtubules (simultaneous polymerization at one end and depolymerization at the other) is observed *in vivo* and *in vitro* at rates of ~1 μm/hr.[141] Similarly, the replisome (the DNA replication machine composed of multiple proteins) continuously replaces the polymerase component to maintain its performance.[142,143] It is an interesting challenge to implement such an approach to self-healing in an engineered nanoscale structure.

In addition to fatigue, the lifetime of molecular machines can be limited by wear and oxidative damage. Unique mechanisms of lubrication may be exploited in molecular machines[144], not only to reduce friction[18], but possibly also to counteract wear. Reactions with reactive oxygen species are of course a major cause of protein degradation, and mechanical activation of bonds may further accelerate it. At the macroscale, the interaction of corrosion and stress leads to the well-known phenomenon of stress corrosion cracking.

Performance of molecular motors

A prediction regarding the achievable performance of molecular motors can be based on recently discovered universal scaling laws for force generation by linear and rotary motors.[145] The force output of linear motors was found to increase with mass to the 2/3, with an excellent correlation over more than 25 orders of magnitude in mass **(Figure 46)**. The force output of

individual motors deviates from the trendline by less than an order of magnitude in either direction. This fact is not entirely surprising considering that strength vs. density diagrams of materials reveal a similarly narrow distribution around the average,[146] meaning that limited material choices result in limited variations in optimized performance. Extrapolation of the trendline to the molecular scale however reveals that rotaxane motors utilized for artificial muscles[134] have a force output which exceeds the expected output by nearly a hundredfold. It seems likely that increasing the lifetime of these synthetic molecular motors will either require a significant increase in their bulk or a reduction in force. Reducing the force output is not straightforward, since it will be detrimental to efficiency or require the use of a less energetic quantum of input energy (a photon or a fuel molecule). Biomolecular motors use ATP molecules as low-energy fuel and thereby optimize concurrently efficiency[147], size and force.

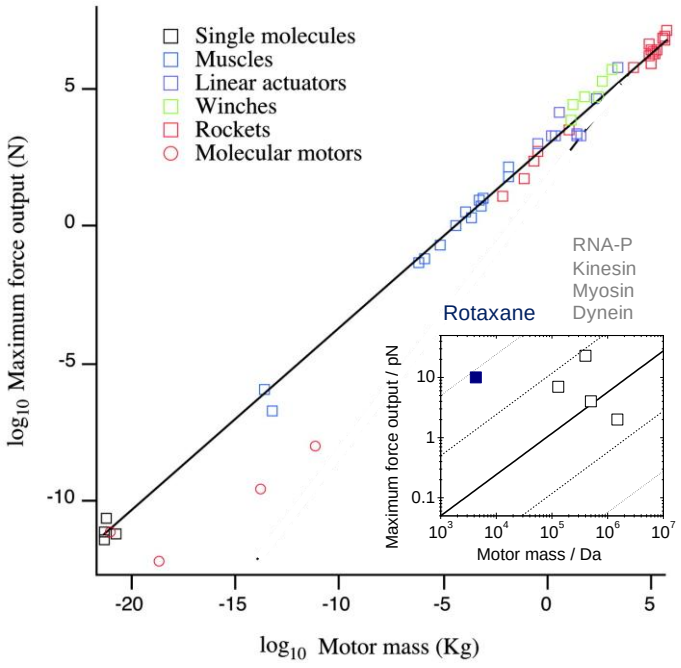

Figure 46. Empirical relation between maximum force of motors as function of their mass as discovered by Marden and Allen (Figure adapted from [145]). The inset extrapolates the trend to the molecular scale. The rotaxane employed for artificial muscles is significantly stronger than

what would be expected from this universal scaling law, which potentially accounts for its small lifetime.[134]

Conclusion

Clearly, the basic considerations presented here raise questions rather than provide answers. However, the addition of an engineering perspective to the prevailing chemical and biophysical viewpoints seems timely and potentially illuminating. Studies of the mechanical fatigue of repetitively stretched titin molecules[148], and the mechanical response of virus shells to repeated deformation[149] provide first insights into the complexity of the response of molecular structures to repeated mechanical stresses.

Proof of equations (3).

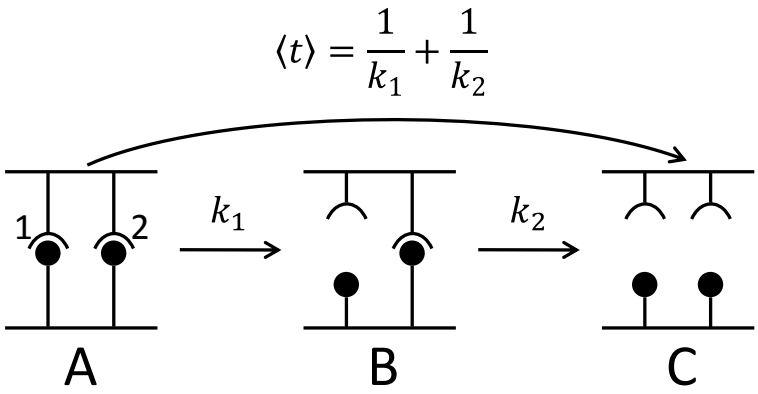

Figure 47. A system with two molecular bonds without rebinding.

The average time of the formation of the C state is given by:

$$\langle t \rangle = \frac{\int_0^\infty t \frac{dC}{dt} dt}{\int_0^\infty \frac{dC}{dt} dt}$$

The equations governing the system are the following:

$$\begin{cases} \dfrac{dA}{dt} = -k_1 A \\ \dfrac{dB}{dt} = k_1 A - k_2 B \\ \dfrac{dC}{dt} = k_2 B \\ A + B + C = A_0 \\ B(t=0) = 0 \\ C(t=0) = 0 \end{cases}$$

We then obtain:

$$A = A_0 e^{-k_1 t}$$

When deriving the second equation of the system and using the expression of A, we obtain:

$$\frac{d^2 B}{dt^2} + k_2 \frac{dB}{dt} = -k_1^2 A_0 e^{-k_1 t}$$

A particular solution of this equation is under the form $dB/dt = \beta e^{-k_1 t}$ and the homogenous solution under the form $dB/dt = \alpha e^{-k_2 t}$. Plugging the particular solution in the 2nd order differential equation of dB/dt, we obtain $\beta = -k_1^2 A_0/(k_2 - k_1)$. Now all the solutions for dB/dt are under the form $\alpha e^{-k_2 t} - k_1 A_0/(k_1 - k_2)\, e^{-k_1 t}$. Using the initial condition $dB/dt\,(t=0) = k_1 A_0$, we obtain the expression for dB/dt:

$$\frac{dB}{dt} = \frac{k_1 A_0}{k_2 - k_1}(-k_1 e^{-k_1 t} + k_2 e^{-k_2 t})$$

By integrating and using the initial condition on B, we obtain:

$$\frac{dC}{dt} = \frac{k_1 k_2 A_0}{k_2 - k_1}(e^{-k_1 t} - e^{-k_2 t})$$

By simply integrating this equation between zero and infinity, we obtain:

$$\int_0^\infty \frac{dC}{dt} dt = A_0$$

On the other hand, we have:

$$\int_0^\infty t \frac{dC}{dt} dt = \frac{k_1 k_2 A_0}{k_1 - k_2} \left(- \int_0^\infty t e^{-k_1 t} dt + \int_0^\infty t e^{-k_2 t} dt \right)$$

We integrate by parts both integrals and obtain:

$$\int_0^\infty t \frac{dC}{dt} dt = \frac{k_1 k_2 A_0}{k_1 - k_2} \left(- \int_0^\infty \frac{1}{k_1} e^{-k_1 t} dt + \int_0^\infty \frac{1}{k_2} e^{-k_2 t} dt \right)$$

Finishing the calculation, we obtain:

$$\int_0^\infty t \frac{dC}{dt} dt = \frac{k_1 + k_2}{k_1 k_2} A_0$$

To conclude, we obtain the average time:

$$\langle t \rangle = \frac{1}{k_1} + \frac{1}{k_2}$$

Proof of equation (4).

$$\langle t \rangle = \frac{k_1 + k_{-1} + k_2}{k_1 k_2}$$

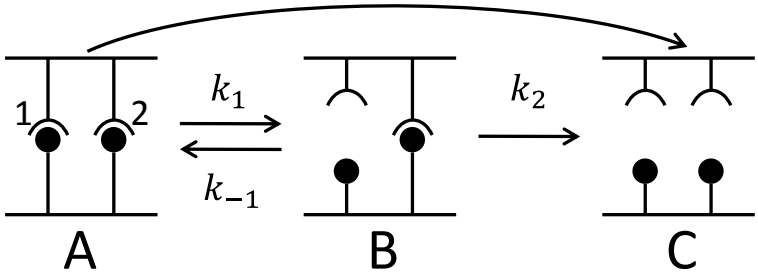

Figure 48. A system with two molecular bonds with rebinding.

The average time of the formation of the C state is given by:

$$\langle t \rangle = \frac{\int_0^\infty t \frac{dC}{dt} dt}{\int_0^\infty \frac{dC}{dt} dt}$$

The equations governing the system are the following:

$$\begin{cases} \frac{dA}{dt} = -k_1 A + k_{-1} B \\ \frac{dB}{dt} = k_1 A - (k_2 + k_{-1}) B \\ \frac{dC}{dt} = k_2 B \\ A + B + C = A_0 \\ B(t = 0) = 0 \\ C(t = 0) = 0 \end{cases}$$

Re-writing the system with matrices, we have $\dot{X} = SX$ where:

$$S = \begin{bmatrix} -k_1 & k_{-1} & 0 \\ k_1 & -(k_2 + k_{-1}) & 0 \\ 0 & k_2 & 0 \end{bmatrix}$$

And:

$$X = \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$

The rank of S is 2 and we are looking for its eigenvalues by solving $\det(A - \lambda I) = 0$ where λ is real number and where I is the identity matrix in 3x3.

$$\det(A - \lambda I) = -\lambda[\lambda^2 + (k_1 + k_{-1} + k_2)\lambda + k_1 k_2]$$

So we have 3 eigenvalues: 0, λ_1, and λ_2 where:

$$\lambda_{1,2} = -\frac{1}{2}(k_1 + k_{-1} + k_2) \pm \frac{1}{2}\sqrt{\Delta}$$

Where:

$$\Delta = (k_1 + k_{-1} + k_2)^2 - 4k_1 k_2 \geq 0$$

Now we find the eigenvectors associated to the eigenvalues 0, λ_1, and λ_2. For the eigenvalue 0:

$$0 \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} = S \times \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

We find the vector $X_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$. For the eigenvalues λ_1 and λ_2:

$$\lambda_{1,2} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} = S \times \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Resolving the two systems, we find the eigenvector X_{λ_1} for the eigenvalue λ_1:

$$X_{\lambda_1}=\begin{bmatrix} \frac{k_{-1}\lambda_1}{k_1+k_2} \\ \lambda_1 \\ \\ k_2 \end{bmatrix}$$

And we find the eigenvector X_{λ_2} for the eigenvalue λ_2:

$$X_{\lambda_2}=\begin{bmatrix} \frac{k_{-1}\lambda_2}{k_1+k_2} \\ \lambda_2 \\ \\ k_2 \end{bmatrix}$$

We introduce the matrix T formed from the eigenvectors:

$$T=\begin{bmatrix} 0 & \frac{k_{-1}\lambda_1}{k_1+k_2} & \frac{k_{-1}\lambda_2}{k_1+k_2} \\ 0 & \lambda_1 & \lambda_2 \\ \\ 1 & k_2 & k_2 \end{bmatrix}$$

By introducing the vector Y:

$$Y=\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Such as $X = SY$. Replacing $X = SY$ in $\dot{X} = SX$**,** we obtain:

$$\dot{Y} = DY$$

With:

$$D=\begin{bmatrix} 0 & 0 & 0 \\ 0 & \lambda_1 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix}$$

Solving $\dot{Y} = DY$, we obtain:

$$\begin{cases} y_1 = Y_1 \\ y_2 = Y_2 e^{\lambda_1 t} \\ y_3 = Y_3 e^{\lambda_2 t} \end{cases}$$

Where $Y_1, Y_2,$ and Y_3 are real numbers. Using $X = SY$, we obtain:

$$C = Y_1 + k_2\left(Y_2 e^{\lambda_1 t} + Y_3 e^{\lambda_2 t}\right)$$

When solving for B and using the initial condition $B(t = 0) = 0$, we obtain:

$$Y_2 = -\frac{\lambda_2}{\lambda_1} Y_3$$

We then have:

$$\int_0^\infty \frac{dC}{dt}\, dt = -k_2(Y_2 + Y_3)$$

And:

$$\int_0^\infty t\frac{dC}{dt}\, dt = \frac{k_2 Y_2}{\lambda_1} + \frac{k_2 Y_3}{\lambda_2}$$

By re-arranging and using $Y_2 = -\lambda_2/\lambda_1 Y_3$, I obtain:

$$\langle t \rangle = \frac{k_1 + k_2 + k_{-1}}{k_1 k_2}$$

CHAPTER 6

Random Sequential Adsorption of proteins on polymer-covered surfaces: A simulation-based approach

A thing of beauty is a joy forever

John Keats

In preparation as E.L.P. Dumont, A. Guillaume, A. Gore, and H. Hess, *Random Sequential Adsorption of proteins on polymer-covered surfaces: A simulation-based approach.*

Introduction

Non-fouling polymeric coatings enable the suppression of protein adsorption to surfaces, and their perfection is the objective of many recent experimental studies.[20-25] Obtaining a theoretical understanding of the functioning of these coatings and the prediction of residual protein adsorption as a function of the coating properties has similarly attracted significant interest.[26-35] Our recent study developed a basic Random Sequential Adsorption (RSA) model for protein adsorption to non-fouling coatings, which was analytically solvable and yielded encouraging agreement with published experimental data.[36] The model assumed that polymer chains on the surface can be represented by hard spheres with a radius equal to their radius of gyration. These randomly distributed hard spheres obstruct the adsorption of proteins, again represented as hard spheres with a diameter equal to the diameter of the protein. The evolution of the protein density on the surface was calculated from the independent probabilities to penetrate the layer of adsorbed proteins and the layer of polymer chains.

Simulations of random sequential adsorption

Random sequential adsorption models are well suited to reflect two aspects of the protein adsorption process.[150,151] Many experiments detect only tightly bound proteins, either because protein which are not absorbed are washed off or because the detection method is not sensitive to weakly bound proteins. As a result, bound proteins can be considered to be irreversibly adsorbed. Secondly, an already adsorbed protein blocks the adsorption of further proteins so that the protein adsorption problem is similar to a “parking problem”.

Simulations facilitate the study of complex random sequential adsorption processes. These include, for example, adsorption of a binary mixture of disks,[152] changes in particle dimensions

after adsorption,[153] asymmetric particles adsorbing in different configurations,[154] and particles adsorbing to a surface covered with obstacles as described here.[36]

In our simulations, a square lattice with 4,000 x 4,000 lines represents a 100 nm x 100 nm surface area. The lattice spacing of 25 pm (1/10 of the diameter of an atom) is sufficiently small since it is smaller than the uncertainty on the location of the particles due to thermal fluctuations, It is also significantly smaller than polymer chains or proteins. Polymer chains are represented by spherical particles and are first placed on the surface in a random sequential adsorption process. For this first step, the particle radius, designed as r_{hard} is calculated from the experimentally determined[32] maximum grafting density ρ_o according to:

$$\rho_o \pi {r_{\mathrm{hard}}}^2 = 0.546$$

Where the value 0.546 corresponds to the surface coverage by the polymers at the jamming limit (see the section "Results"). The calculation of r_{hard} relies on the assumption that the polymer chains on a surface use the maximum surface available to them in a random placing.

The polymer placement process is terminated when the coverage x defined as a fraction of the maximum grafting density is reached. The coverage x is defined by:

$$x = \frac{N_{\mathrm{poly}}}{\rho_o L^2}$$

Where N_{poly} is the number of placed polymers and $L = 100\,\mathrm{nm}$ is the side length of the simulated surface area.

After the polymer chains have been distributed, the area available to protein adsorption is determined. Each polymer chain obstructs a circular area with radius r_{min} given by:

$$r_{\min} = 2\sqrt{r_{\text{poly}} r_{\text{prot}}}$$

Where r_{poly} is the effective radius of the polymer, defined as the radius of gyration, and r_{prot} is the radius of the adsorbing protein, again assumed to be a spherical particle (**Figure 49**). The calculation of the radius of gyration for the polymers depends on the length of their chains. For $n > 12$, n being the degree of polymerization, r_{poly} is given by ${r_{\text{poly}}}^2 = nbl_k \cos(\Psi)/6$ and by ${r_{\text{poly}}}^2 = nb^2/6$ for smaller chains.[36] The Kuhn length of the polymer l_k is here taken as 1 nm, the monomer length b is 0.278 nm, and the angle between the chain axis and the bond between monomers Ψ is 37.5°.

A protein is placed into the available area, and the obstructed circular area with a radius of twice the protein radius is removed from the available area. This is repeated until no area remains available. The result of this procedure is the available area $A(x, N_{\text{poly}}, r_{\text{poly}}, r_{\text{prot}})$ as a function of the number of proteins adsorbed and the maximum number of proteins which can be adsorbed to this particular distribution of polymer chain locations.

The kinetics of the adsorption process can be obtained by considering that each placement of a protein into the available area has to be accompanied by a number of unsuccessful attempts which, in average, is proportional to the ratio between the unavailable area and the available area. Thus, the time necessary to adsorb N proteins is defined by:

$$t(N) = \sum_{i=0}^{N-1} \frac{1}{SZA(i)}$$

Where S is the sticking probability for a protein-surface collision and Z is the rate of collision between proteins and surface per unit of area. To compare our results with Katira et al.[36], S is set to 8.8×10^{-8}.[155] The collision rate Z is given by:

$$Z = C_p * \sqrt{\frac{k_b T}{m_p}}$$

Where C_p is the protein concentration and m_p is the protein mass.[155,156]

In contrast to Feder's method where many attempts to place a protein have to be made as the surface coverage increases,[150] our method is fast when a large number of polymer chains on the surface reduces the available area far below the area of the entire square, since there are no unsuccessful attempts. This situation applies by definition to non-fouling coatings.

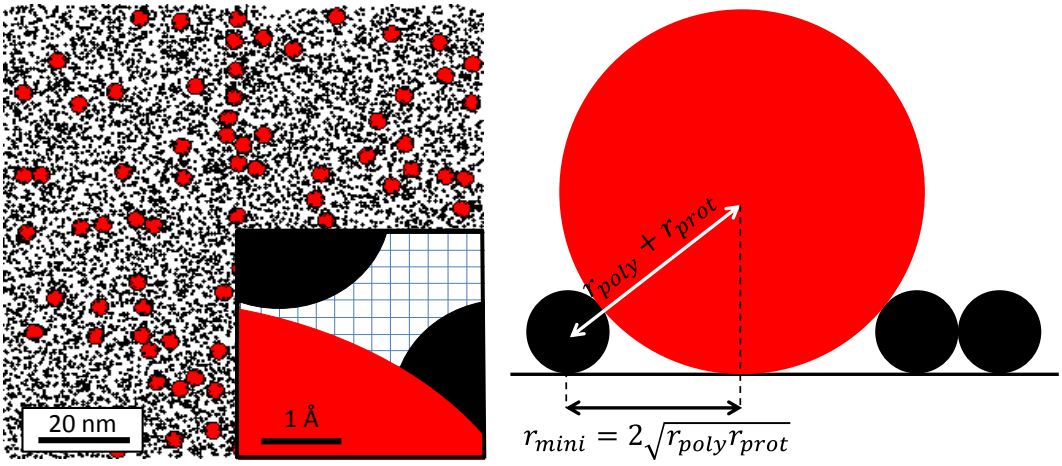

Figure 49. The two stage random sequential adsorption process is conducted on a lattice. First, particles representing polymer chains are placed on the surface (black). Secondly, particles representing proteins (red) are placed in the available spaces between the polymers. The mesh of 25 pm is visible in the inset. Here, protein Ribonuclease A is placed after a polymer coating of EG_6OH at $x = 0.2$ is achieved.

Results

The simulation procedure is first validated for the adsorption of proteins to a bare surface ($r_{\mathrm{prot}} = 1.9\ \mathrm{nm}$ for Ribonuclease A[157]), which corresponds to the previously studied random sequential adsorption of disks. The protein coverage θ is defined as follows:

$$\theta = \frac{N_{\mathrm{prot}} \pi r_{\mathrm{prot}}^2}{L^2}$$

Then the maximum coverage is:

$$\theta_\infty = 0.546 \pm 0.001$$

This result is in agreement with the literature. The error is the standard deviation obtained from five independent simulation runs. Secondly, as previously found by Feder,[150] in the vicinity of the jamming limit the coverage saturation increases in proportion to $t^{-0.5}$ (**Figure 50**):

$$\theta_\infty - \theta \propto \frac{1}{\sqrt{t}}$$

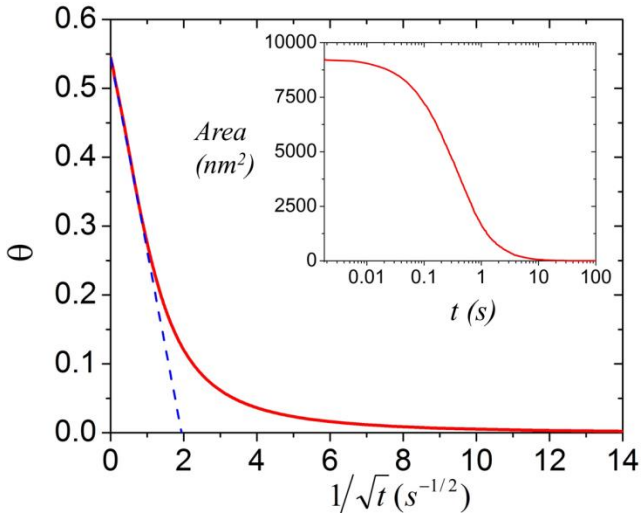

Figure 50. Protein coverage as a function if time. Protein coverage on a bare surface (blue) increases in proportion to $t^{-0.5}$ in the vicinity of the jamming limit (0.546), as shown by the

dotted line. In the inset, the decrease of the area available to new proteins in function of time t. Preliminary results. Simulations by A. Guillaume.

The simulation of protein adsorption ($r_{\mathrm{prot}} = 1.9\ \mathrm{nm}$) to a surface covered with polymers ($r_{\mathrm{poly}} = 0.278\ \mathrm{nm}$ for EG_6OH chains according to Kuhn's formula for $n = 6$) at 10% of the maximal polymer coverage ($x = 0.1$) reveals a maximum number of adsorbed proteins of 355 which corresponds to a coverage of about 40%. We found that the available area depends on the surface density theta θ in the following relationship:

$$A^m = w(\theta_\infty - \theta)$$

Where A is the area in nm^{-2}. Running simulations for other percentages of the maximal polymer coverage ($x = 0.05, x = 0.2, x = 0.4$), and then for polymers of different size ($r_{\mathrm{poly}} = 0.114\ \mathrm{nm}$ for EGOH and $r_{\mathrm{poly}} = 0.79\ \mathrm{nm}$ for $EG_{17}OCH_3$) showed that the coefficient m found is fairly equal to 0.36, when w varies with the polymer size and coverage (**Figure 51**).

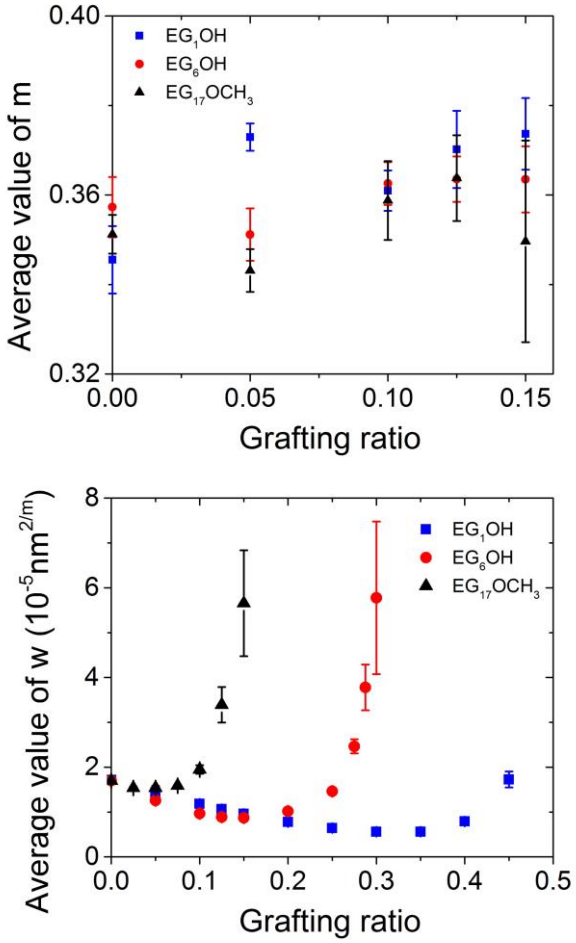

Figure 51. m and w as a function of the grafting ratio for several polymer brushes. Simulations by A. Guillaume and A. Gore.

We then plot the logarithm of the initial area available for protein placement as a function of the polymer grafting ratio. As expected, this initial available area decreases with higher grafting ratio of polymers (**Figure 52**).

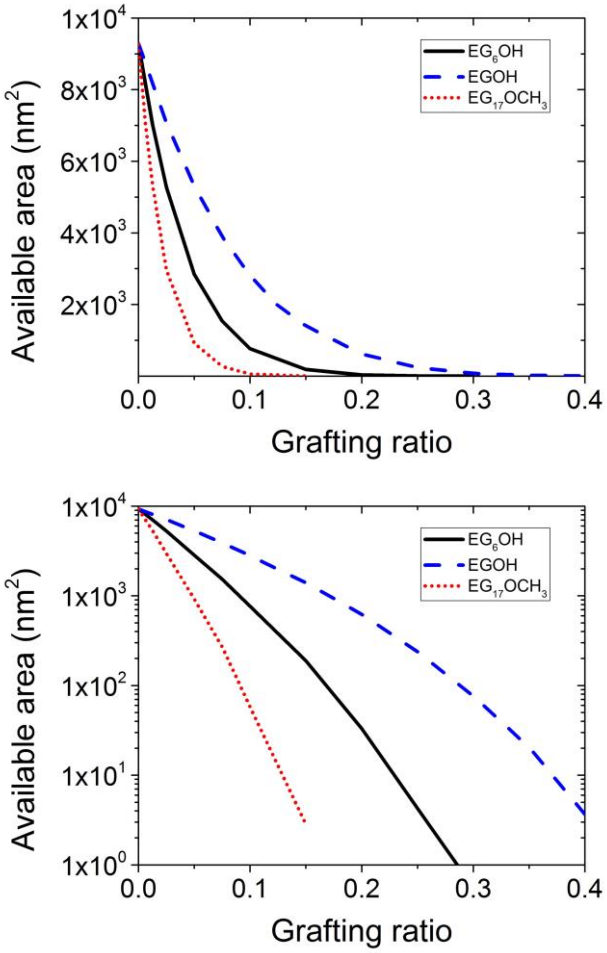

Figure 52. The initial Available Area for proteins decreases when the polymer grafting ratio increases. Logically, it decreases faster when the coated polymer is big ($EG_{17}OCH_3$). Plotting the logarithm of this initial available area shows a linear trend for small grafting ratios. Preliminary simulations. Simulations by A. Guillaume.

These simulations are currently being continued with the goal of developing a new analytical model for random sequential adsorption of proteins on a polymer-grafted surface.

CHAPTER 7

A molecular model to explain material fracture at low stresses

La science est obscure – peut-être parce que la vérité est sombre

(Science is obscure – maybe because the truth is dark)

Victor Hugo

Zhurkov's molecular model to explain fractures of materials

In 1965, Zhurkov introduced a model[37] to predict the lifetime of materials under uniaxial tension. The model is very simple yet it connects τ the lifetime of the material and σ the uniaxial stress (macroscopic experimental values) to the following microscopic constants: τ_0 the inverse of the oscillation frequency of atoms in the solid, U_0 the enthalpy of sublimation, k_B the Boltzmann constant, T the temperature, and v the parameter that measures the disorientation of the molecular structure. The equation proposed by Zhurkov is the following:

$$\tau = \tau_0 \exp\left(\frac{U_0 - v\sigma}{k_B T}\right)$$

This equation accounts very well for the lifetime of several materials as a function of stress. Below, a figure of the paper is reproduced.

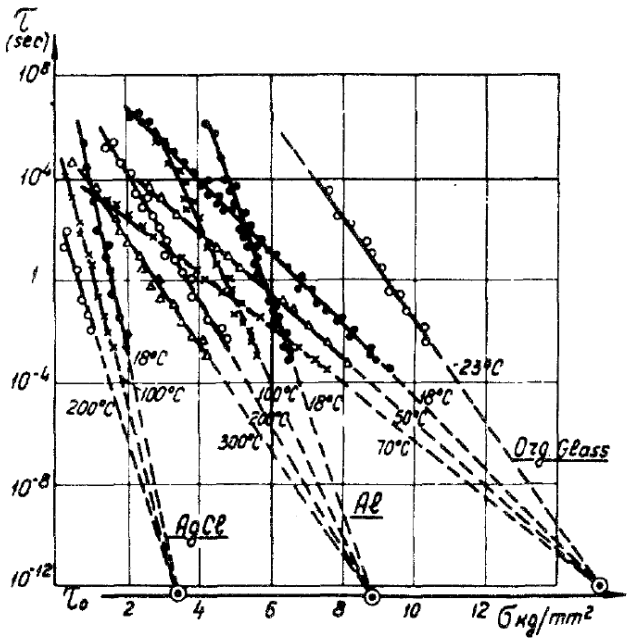

Figure 53. Prediction of the lifetime of silver chloride, aluminum, and Poly(methyl methacrylate) as a function of uniaxial stress at different temperatures. Reproduced from [37].

Model at low stresses

Zhurkov's model does not account for the behavior of fractures at low stresses and high temperatures. Zhurkov stipulates that the deviation comes from the reversibility of the bond ruptures in the solid. From that hypothesis, I built a simple molecular model that accounts for the reversibility of the molecular bond ruptures:

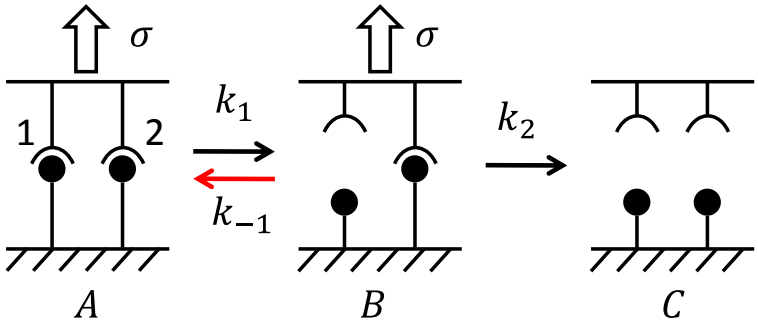

Figure 54. Improvement of Zhurkov's model to account for the behavior at low stresses.

Like in chapter 5, the governing equations are:

$$\begin{cases} \frac{dA}{dt} = -k_1 A + k_{-1} B \\ \frac{dB}{dt} = k_1 A - (k_{-1} + k_2) B \\ \frac{dC}{dt} = k_2 B \\ A + B + C = A_0 \\ B(t = 0) = 0 \\ C(t = 0) = 0 \end{cases}$$

The average time to failure is (from Chapter 5):

$$\tau = \frac{\int_0^\infty t \frac{dC}{dt} dt}{\int_0^\infty \frac{dC}{dt} dt} = \frac{k_1 + k_2 + k_{-1}}{k_1 k_2}$$

I set k_1 and k_2 to be:

$$\begin{cases} k_1 = \frac{1}{\tau_0} \exp\left(-\frac{U_0 - v\sigma}{2k_B T}\right) \\ k_{-1} = \frac{1}{\tau_0} \exp\left(-\frac{U_0{}' + v'\sigma}{k_B T}\right) \\ k_2 = k_1 \end{cases}$$

The atomic volume of a metal is approximately 10^{-29} m^3 and the enthalpy of sublimation approximately 10^{-19} J. I reproduce below the curves from Figure 8 from Zhurkov's paper with the rebinding model:

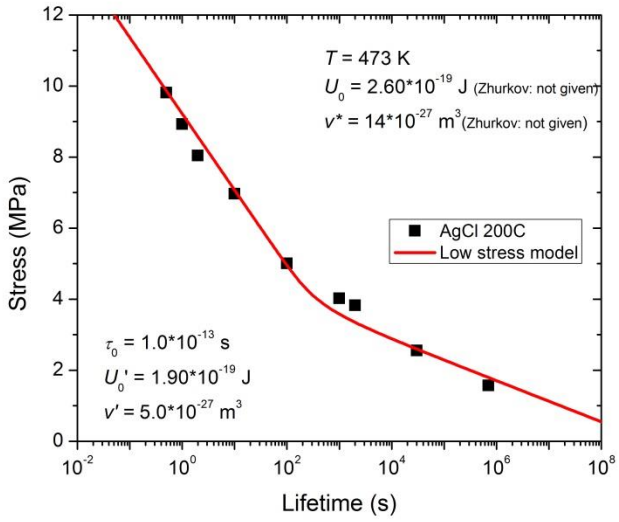

Figure 55. Low stress model for silver chloride at 200 C. Data taken from [37].

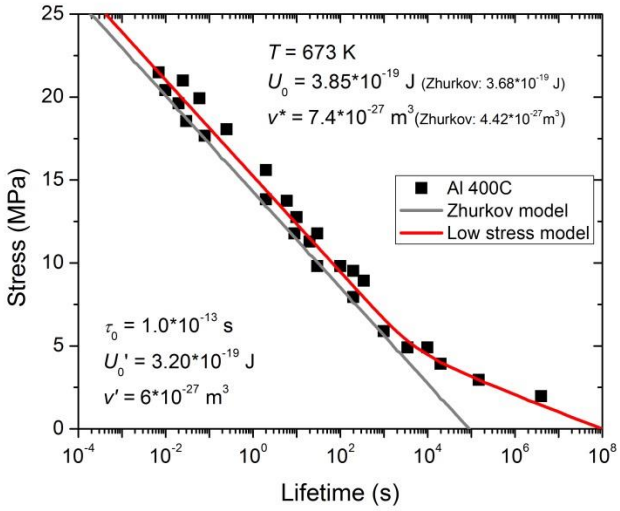

Figure 56. Low stress model for aluminum at 400 C. Data taken from [37].

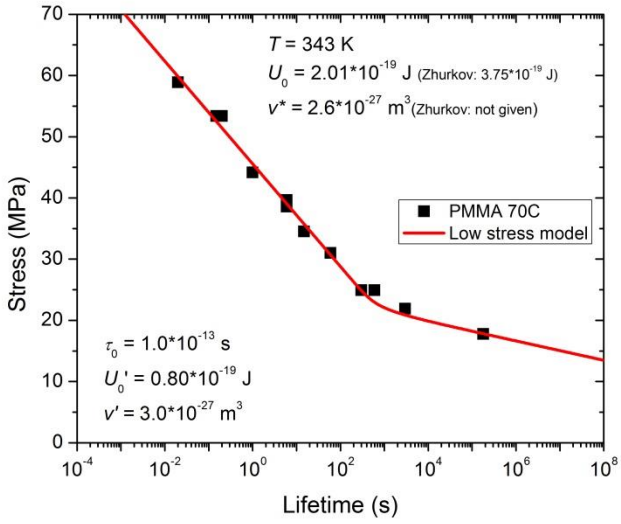

Figure 57. Low stress model for PMMA at 70 C. Data taken from [37].

It is not clear yet what the parameters $U_0{}'$ and v' represent.

MATLAB code for the low stress model

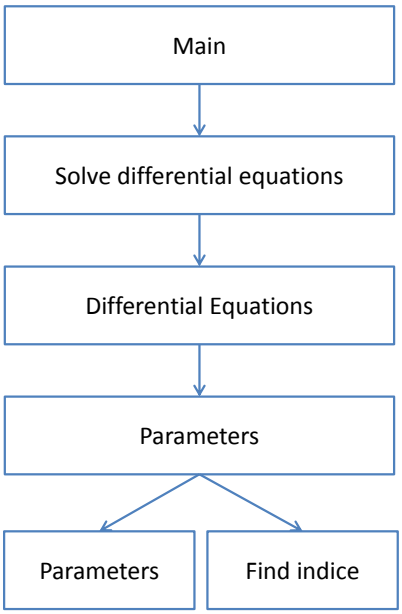

Figure 58. Structure of the MATLAB code for the low stress model.

Main.m

```
function [av_lifetimes,force_array] = Main()

%% Chose data set (Fig 8.1, 8.2, or 8.3)

[lifetime_data,force_data]        =       Fig8_2_zhurkov()       ;

% CHANGE VALUES IN PARAMETERS.M WHEN CHANGE THE FIGURE NUMBER

%% Number of bonds in the system

global          n                               ;

n               =               2               ;

%% Parameter of simulations

step            =               0.1             ; % in MPa
inf_force       =               0               ; % in MPa
sup_force       =               25              ; % in MPa
% 12 MPa for Fig 8.1 -- 25 MPa for Fig 8.2 -- 70 MPa for Fig 8.3

number_steps    =               round((sup_force-inf_force)/step);

av_lifetimes    =               zeros(1,number_steps);
force_array     =               zeros(1,number_steps);


for i=1:number_steps+1

    force_temp=inf_force+step*(i-1);
    av_lifetimes(i) = Average_lifetime(force_temp*1e6); % From MPa to Pa!
    force_array(i) = force_temp;

end

%% Plot simulation against data

semilogx(av_lifetimes,force_array,'linewidth',2);
hold on;
```

```
semilogx(lifetime_data,force_data,'square','color','red');
hold on;

xlabel('Average time of failure (s)','fontsize',20);
ylabel('Stress (MPa)','fontsize',20);

% To copy/paste the values into ORIGIN
force_array = force_array';
av_lifetimes = av_lifetimes';



end
```

Average_lifetime.m

```
function [average_time] = Average_lifetime(f)

global force;
force                   =    f                              ;

global n;

disp(['Force is ',num2str(force)]);

[T,Y]                   =    Solve_differential_equations();

rate_product_formation =    diff(Y(:,n+1))./diff(T);


T_resized = zeros(length(T)-1,1);
for i=1:length(T)-1
    T_resized(i) = T(i);
end

numerator_av_time = sum(T_resized.*rate_product_formation.*diff(T));
denominator_av_time = Y(length(T),n+1)-Y(1,n+1);

average_time = numerator_av_time./denominator_av_time;

end
```

Solve_differential_equations.m

```
function [T,Y] = Solve_differential_equations()

global n;

global max_time;
max_time = 1e-50;


initial_conditions = zeros(n+1,1);
initial_conditions(1) = 1;

% Maximum time over which the equations are solved.
disp(['Maximum time is ',num2str(max_time)]);

criteria = false;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%criteria = true;
%[T,Y] = ode15s(@Differential_equations,[0 1e-1],initial_conditions);
%plot(T,Y);
%%plot(T,Y(:,n+1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

while criteria == false
    [T,Y] = ode15s(@Differential_equations,[0 max_time],initial_conditions);
    indice = Find_indice(T);
    if Y(indice,n+1)>0.5
        criteria = true;
    elseif Y(indice,n+1) < 0.5
        max_time = 10*max_time;

    else
```

```
        max_time = 0.5*max_time;
end

end

disp(['Maximum time found is ',num2str(max_time)]);
disp(T(indice));

%plot(T,Y);

end
```

Differential_equations.m

```
function dy = Differential_equations(t,y)

global n;
global force;

%force = 100e4;

[T_0,U_0,x,T_q,U_q,x_q,T] = Parameters();

k = 1.38e-23; % Boltzman constant in J.K-1


if n==1
    A = zeros(2,2);
    p_1 = 1./T_0.*exp(-U_0/(k*T)+x*force/(k*T));
    A(1,1) = -p_1;
    A(2,1) = p_1;
elseif n==2
    A = zeros(3,3);
    % 2nd model and 4th model
    %p_1=1./T_0.*exp(-U_0/(k*T)+x*force/(2*k*T));

    % 3rd mdoel
    p_1=1./T_0.*exp(-U_0/(k*T)+x*force/(k*T));

    p_2=1./T_0.*exp(-U_0/(k*T)+x*force/(k*T));
    q_1= 1/T_0*exp(-U_q/(k*T)-force*x_q./(k*T));
    A(1,1) = -p_1;
    A(1,2) = q_1;
    A(2,1) = p_1;
    A(2,2) = -(q_1+p_2);
    A(3,2) = p_2;
elseif n>2
    A = Matrix_of_coefficients();
end

%disp('The matrix of coefficients');
%disp(A);

dy = zeros(n+1,1);

dy = A*y;


end
```

Find_indice.m

```
function indice = Find_indice(T)

threshold = 0.2*T(length(T));

condition = false;
i=1;

while condition == false
    if T(i)<=threshold && T(i+1)>threshold
        indice = i;
        condition = true;
    end
if i==length(T)-1
```

```
    condition = true;
    indice = i;
end
i = i+1;
end


end
```

Parameters.m

```
function [T_0,U_0,x,T_q,U_q,x_q,T] = Parameters()

%% Zhurkov parameters

T_0         =       1e-13           ; %Period for the bond to unbind
U_0         =       3.85e-19        ; % in Joules;
x           =       7.4e-27         ; % Volume in m3

%% Unbinding parameters

T_q         =       T_0             ; %60e0; % Period for the bond to rebind in s
U_q         =       3.2e-19         ; % In Joule for the rebinding
x_q         =       6e-27           ; % in m3 for the rebinding

%% Tempearture in Kelvin

T           =       400+273.15      ;

end
```

References

1 Déjardin, P. *Proteins at solid-liquid interfaces*. (Springer, 2006).

2 Ratner, B. D. *Biomaterials science : an introduction to materials in medicine*. 2nd edn, (Elsevier Academic Press, 2004).

3 Grosberg, A., Alford, P. W., McCain, M. L. & Parker, K. K. Ensembles of engineered cardiac tissues for physiological and pharmacological study: Heart on a chip. *Lab Chip* **11**, 4165-4173 (2011).

4 Huh, D. *et al.* Reconstituting Organ-Level Lung Functions on a Chip. *Science* **328**, 1662-1668 (2010).

5 Bhushan, B. *Springer handbook of nanotechnology*. 3rd rev. and extended edn, (Springer, 2010).

6 Dienwiebel, M. & Frenken, J. W. M. *Fundamentals of Friction and Wear on the Nanoscale*. (Springer, 2007).

7 Bhushan, B. (John Wiley & Sons Inc.,, Hoboken, 2013).

8 Ludema, K. C. *Friction, wear, lubrication: A textbook in tribology*. (CRC Press, 1996).

9 Butt, H.-J. r., Graf, K. & Kappl, M. *Physics and chemistry of interfaces*. 2nd., rev. and enl. edn, (Wiley-VCH, 2006).

10 Hutchings, I. M. *Tribology: Friction and wear of engineering materials*. (CRC Press, 1992).

11 Bhushan, B. Nanotribology and nanomechanics in nano/biotechnology. *Philos T R Soc A* **366**, 1499-1537 (2008).

12 Fischer, T., Agarwal, A. & Hess, H. A smart dust biosensor powered by kinesin motors. *Nat Nanotechnol* **4**, 162-166 (2009).

13 De Gennes, P. G. Conformations of Polymers Attached to an Interface. *Macromolecules* **13**, 1069-1075 (1980).

14 Norton, R. L. *Machine design: An integrated approach*. 4th edn, (Prentice Hall, 2011).

15 Erickson, G. M. *et al.* Complex Dental Structure and Wear Biomechanics in Hadrosaurid Dinosaurs. *Science* **338**, 98-101 (2012).

16 Gao, G. T., Mikulski, P. T. & Harrison, J. A. Molecular-scale tribology of amorphous carbon coatings: Effects of film thickness, adhesion, and long-range interactions. *J Am Chem Soc* **124**, 7202-7209 (2002).

17 Jacobs, T. D. B. & Carpick, R. W. Nanoscale wear as a stress-assisted chemical reaction. *Nat Nanotechnol* **8**, 108-112 (2013).

18 Bormuth, V., Varga, V., Howard, J. & Schaffer, E. Protein Friction Limits Diffusive and Directed Movements of Kinesin Motors on Microtubules. *Science* **325**, 870-873 (2009).

19 Norton, R. L. *Machine Design: An Integrated Approach*. (Prentice Hall, 2000).

20 Hucknall, A., Rangarajan, S. & Chilkoti, A. In Pursuit of Zero: Polymer Brushes that Resist the Adsorption of Proteins. *Adv Mater* **21**, 2441-2446 (2009).

21 Zhang, Z. *et al.* Blood compatibility of surfaces with superlow protein adsorption. *Biomaterials* **29**, 4285-4291 (2008).

22 Jiang, S. Y. & Cao, Z. Q. Ultralow-Fouling, Functionalizable, and Hydrolyzable Zwitterionic Materials and Their Derivatives for Biological Applications. *Adv Mater* **22**, 920-932 (2010).

23 Gon, S., Bendersky, M., Ross, J. L. & Santore, M. M. Manipulating Protein Adsorption using a Patchy Protein-Resistant Brush. *Langmuir* **26**, 12147-12154 (2010).

24 Ionov, L., Synytska, A., Kaul, E. & Diez, S. Protein-Resistant Polymer Coatings Based on Surface-Adsorbed Poly(aminoethyl methacrylate)/Poly(ethylene glycol) Copolymers. *Biomacromolecules* **11**, 233-237 (2010).

25 Estephan, Z. G., Schlenoff, P. S. & Schlenoff, J. B. Zwitteration As an Alternative to PEGylation. *Langmuir* **27**, 6794-6800 (2011).

26 Szleifer, I. Polymers and proteins: Interactions at interfaces. *Current Opinion in Solid State & Materials Science* **2**, 337-344 (1997).

27 Morra, M. On the molecular basis of fouling resistance. *Journal of Biomaterials Science-Polymer Edition* **11**, 547-569 (2000).

28 Latour, R. A. in *Encyclopedia of Biomaterials and Biomedical Engineering* (eds G. Wnek & G. Bowlin) (Taylor&Francis, 2005).

29 Jeon, S. I., Lee, J. H., Andrade, J. D. & Degennes, P. G. Protein Surface Interactions in the Presence of Polyethylene Oxide .1. Simplified Theory. *J Colloid Interf Sci* **142**, 149-158 (1991).

30 Jeon, S. I. & Andrade, J. D. Protein Surface Interactions in the Presence of Polyethylene Oxide .2. Effect of Protein Size. *J Colloid Interf Sci* **142**, 159-166 (1991).

31 Szleifer, I. Protein adsorption on surfaces with grafted polymers: A theoretical approach. *Biophys J* **72**, 595-612 (1997).

32 Satulovsky, J., Carignano, M. A. & Szleifer, I. Kinetic and thermodynamic control of protein adsorption. *P Natl Acad Sci USA* **97**, 9037-9041 (2000).

33 Agashe, M., Raut, V., Stuart, S. J. & Latour, R. A. Molecular simulation to characterize the adsorption behavior of a fibrinogen gamma-chain fragment. *Langmuir* **21**, 1103-1117 (2005).

34 Hower, J. C., He, Y., Bernards, M. T. & Jiang, S. Y. Understanding the nonfouling mechanism of surfaces through molecular simulations of sugar-based self-assembled monolayers. *Journal of Chemical Physics* **125**, 214704 (2006).

35 Jiang, S. Y. *et al.* Molecular simulation studies of protein interactions with zwitterionic phosphorylcholine self-assembled monolayers in the presence of water. *Langmuir* **24**, 10358-10364 (2008).

36 Katira, P., Agarwal, A. & Hess, H. A Random Sequential Adsorption Model for Protein Adsorption to Surfaces Functionalized with Poly(ethylene oxide). *Adv Mater* **21**, 1599-1604 (2009).

37 Zhurkov, S. N. Kinetic Concept of the Strength of Solids. *Int J Fracture* **26**, 295-307 (1984).

38 Alberts, B. *Molecular biology of the cell*. 5th edition. edn, (Garland Science, 2008).

39 Howard, J. *Mechanics of motor proteins and the cytoskeleton*. (Sinauer Associates, Publishers, 2001).

40 Vale, R. D., Reese, T. S. & Sheetz, M. P. Identification of a Novel Force-Generating Protein, Kinesin, Involved in Microtubule-Based Motility. *Cell* **42**, 39-50 (1985).

41 Varga, V., Leduc, C., Bormuth, V., Diez, S. & Howard, J. Kinesin-8 Motors Act Cooperatively to Mediate Length-Dependent Microtubule Depolymerization. *Cell* **138**, 1174-1183 (2009).

42 Gigant, B. *et al.* Structure of a kinesin-tubulin complex and implications for kinesin motility. *Nat Struct Mol Biol* **20**, 1001-+ (2013).

43 Cochran, J. C. & Kull, F. J. A molecular motor finds its track. *Nat Struct Mol Biol* **20**, 920-921 (2013).

44 Coy, D. L., Wagenbach, M. & Howard, J. Kinesin takes one 8-nm step for each ATP that it hydrolyzes. *J Biol Chem* **274**, 3667-3671 (1999).

45 Hua, W., Chung, J. & Gelles, J. Distinguishing inchworm and hand-over-hand processive kinesin movement by neck rotation measurements. *Science* **295**, 844-848 (2002).

46 Yildiz, A., Tomishige, M., Vale, R. D. & Selvin, P. R. Kinesin walks hand over hand. *Biophys J* **86**, 527a-527a (2004).

47 Vale, R. D., Schnapp, B. J., Reese, T. S. & Sheetz, M. P. Organelle, Bead, and Microtubule Translocations Promoted by Soluble Factors from the Squid Giant-Axon. *Cell* **40**, 559-569 (1985).

48 Lodish, H. F. *Molecular cell biology*. 4th edn, (W.H. Freeman, 2000).

49 Wilson, L. *et al.* Taxol Stabilization of Microtubules Invitro - Dynamics of Tubulin Addition and Loss at Opposite Microtubule Ends. *Biochemistry-Us* **24**, 5254-5262 (1985).

50 Schiff, P. B., Fant, J. & Horwitz, S. B. Promotion of Microtubule Assembly Invitro by Taxol. *Nature* **277**, 665-667 (1979).

51 Howard, J., Hudspeth, A. J. & Vale, R. D. Movement of Microtubules by Single Kinesin Molecules. *Nature* **342**, 154-158 (1989).

52 Brunner, C., Ernst, K. H., Hess, H. & Vogel, V. Lifetime of biomolecules in polymer-based hybrid nanodevices. *Nanotechnology* **15**, S540-S548 (2004).

53 Ruhnow, F., Zwicker, D. & Diez, S. Tracking Single Particles and Elongated Filaments with Nanometer Precision. *Biophys J* **100**, 2820-2828 (2011).

54 Cohen, F. E. & Kelly, J. W. Therapeutic approaches to protein-misfolding diseases. *Nature* **426**, 905-909 (2003).

55 Leduc, C. *et al.* Molecular crowding creates traffic jams of kinesin motors on microtubules. *P Natl Acad Sci USA* **109**, 6100-6105 (2012).

56 Alexander, S. Adsorption of Chain Molecules with a Polar Head a-Scaling Description. *J Phys-Paris* **38**, 983-987 (1977).

57 Brittain, W. J. & Minko, S. A structural definition of polymer brushes. *J Polym Sci Pol Chem* **45**, 3505-3512 (2007).

58 Milner, S. T. Polymer Brushes. *Science* **251**, 905-914 (1991).

59 Wu, T., Efimenko, K. & Genzer, J. Combinatorial study of the mushroom-to-brush crossover in surface anchored polyacrylamide. *J Am Chem Soc* **124**, 9394-9395 (2002).

60 Wittmer, J., Johner, A., Joanny, J. F. & Binder, K. Chain Desorption from a Semidilute Polymer Brush - a Monte-Carlo Simulation. *J Chem Phys* **101**, 4379-4390 (1994).

61 Jost, P. Tribology: How a word was coined 40 years ago. *Tribol Lubr Technol* **62**, 24-28 (2006).

62 Archard, J. F. Contact and Rubbing of Flat Surfaces. *J Appl Phys* **24**, 981-988 (1953).

63 Meyer, E. *Fundamentals of friction and wear*. 1st edn, (Berlin Heidelberg, 2006).

64 Bhushan, B. *Introduction to tribology*. (John Wiley & Sons, 2002).

65 Binder, K. & Milchev, A. Polymer brushes on flat and curved surfaces: How computer simulations can help to test theories and to interpret experiments. *J Polym Sci Pol Phys* **50**, 1515-1555 (2012).

66 Yamamoto, S., Ejaz, M., Tsujii, Y. & Fukuda, T. Surface interaction forces of well-defined, high-density polymer brushes studied by atomic force microscopy. 2. Effect of graft density. *Macromolecules* **33**, 5608-5612 (2000).

67 Factor, B. J., Lee, L. T., Kent, M. S. & Rondelez, F. Observation of Chain Stretching in Langmuir Diblock Copolymer Monolayers. *Phys Rev E* **48**, R2354-R2357 (1993).

68 Kent, M. S., Lee, L. T., Factor, B. J., Rondelez, F. & Smith, G. S. Tethered Chains in Good Solvent Conditions - an Experimental-Study Involving Langmuir Diblock Copolymer Monolayers. *J Chem Phys* **103**, 2320-2342 (1995).

69 Bright, J. N., Hoh, J. H. & Woolf, T. B. Computational investigation of confined unstructured proteins. *Biophys J* **80**, 407a-408a (2001).

70 Lele, T. P. & Kumar, S. Brushes, cables, and anchors: Recent insights into multiscale assembly and mechanics of cellular structural networks. *Cell Biochem Biophys* **47**, 348-360 (2007).

71 Mukhopadhyay, R. & Hoh, J. H. AFM force measurements on microtubule-associated proteins: the projection domain exerts a long-range repulsive force. *Febs Lett* **505**, 374-378 (2001).

72 Lawrence, C. J. *et al.* A standardized kinesin nomenclature. *J Cell Biol* **167**, 19-22 (2004).

73 Coy, D. L., Hancock, W. O., Wagenbach, M. & Howard, J. Kinesin's tail domain is an inhibitory regulator of the motor domain. *Nat Cell Biol* **1**, 288-292 (1999).

74 Kerssemakers, J., Howard, J., Hess, H. & Diez, S. The distance that kinesin-1 holds its cargo from the microtubule surface measured by fluorescence interference contrast microscopy. *P Natl Acad Sci USA* **103**, 15812-15817 (2006).

75 Ozeki, T. *et al.* Surface-Bound Casein Modulates the Adsorption and Activity of Kinesin on SiO2 Surfaces. *Biophys J* **96**, 3305-3318 (2009).

76 Kerssemakers, J. *et al.* 3D Nanometer Tracking of Motile Microtubules on Reflective Surfaces. *Small* **5**, 1732-1737 (2009).

77 Braun, D. & Fromherz, P. Fluorescence interference-contrast microscopy of cell adhesion on oxidized silicon. *Applied Physics A Materials Science & Processing* **65**, 341-348 (1997).

78 Ionov, L. *et al.* Fast and spatially resolved environmental probing using stimuli-responsive polymer layers and fluorescent nanocrystals. *Adv Mater* **18**, 1453-1457 (2006).

79 Lambacher, A. & Fromherz, P. Fluorescence interference-contrast microscopy on oxidized silicon using a monomolecular dye layer. *Applied Physics A Materials Science & Processing* **63**, 207-216 (1996).

80 Lambacher, A. & Fromherz, P. Luminescence of dye molecules on oxidized silicon and fluorescence interference contrast microscopy of biomembranes. *Biophys J* **19**, 1435-1453 (2002).

81 Flory, P. J. *Principles of Polymer Chemistry*. (Cornell University Press, 1953).

82 Katira, P. *et al.* Quantifying the performance of protein-resisting surfaces at ultra-low protein coverages using kinesin motor proteins as probes. *Adv Mater* **19**, 3171-3176 (2007).

83 Agarwal, A., Luria, E., Deng, X. P., Lahann, J. & Hess, H. Landing Rate Measurements to Detect Fibrinogen Adsorption to Non-fouling Surfaces. *Cellular and Molecular Bioengineering* **5**, 320-326 (2012).

84 Howard, J., Hunt, A. J. & Baek, S. Assay of microtubule movement driven by single kinesin molecules. *Methods Cell Biol* **39**, 137-147 (1993).

85 Ray, S., Meyhofer, E., Milligan, R. A. & Howard, J. Kinesin follows the microtubule's protofilament axis. *J Cell Biol* **121**, 1083-1093 (1993).

86 van den Heuvel, M. G. L. & Dekker, C. Motor proteins at work for nanotechnology. *Science* **317**, 333-336 (2007).

87 Goel, A. & Vogel, V. Harnessing biological motors to engineer systems for nanoscale transport and assembly. *Nat Nanotechnol* **3**, 465-475 (2008).

88 Agarwal, A. & Hess, H. Biomolecular motors at the intersection of nanotechnology and polymer science. *Progress in Polymer Science* **35**, 252-277 (2010).

89 Gillingham, A. K. & Munro, S. Long coiled-coil proteins and membrane traffic. *Bba-Mol Cell Res* **1641**, 71-85 (2003).

90 Wetterma.G, Borglund, E. & Brolin, S. E. A Regenerating System for Studies of Phosphoryl Transfer from ATP. *Anal Biochem* **22**, 211-218 (1968).

91 Agarwal, A., Luria, E., Deng, x., Lahann, J. & Hess, H. Landing Rate Measurements to Detect Fibrinogen Adsorption to Non-fouling Surfaces. *Cellular and Molecular Bioengineering* **5**, 320-326 (2012).

92 Diebolt, J. & Robert, C. P. Estimation of Finite Mixture Distributions through Bayesian Sampling. *J Roy Stat Soc B Met* **56**, 363-375 (1994).

93 Raftery, A. E. Approximate Bayes factors and accounting for model uncertainty in generalised linear models. *Biometrika* **83**, 251-266 (1996).

94 Bensmail, H., Celeux, G., Raftery, A. E. & Robert, C. P. Inference in model-based cluster analysis. *Stat Comput* **7**, 1-10 (1997).

95 Tawada, K. & Sekimoto, K. Protein Friction Exerted by Motor Enzymes through a Weak-Binding Interaction. *J Theor Biol* **150**, 193-200 (1991).

96 Nitta, T. & Hess, H. Dispersion in active transport by kinesin-powered molecular shuttles. *Nano Lett* **5**, 1337-1342 (2005).

97 Diaz-Valencia, J. D. *et al.* Drosophila Katanin-60 Depolymerizes and Severs at Microtubule Defects. *Biophys J* **100**, 2440-2449 (2011).

98 Howard, J. & Hyman, A. A. Preparation of marked microtubules for the assay of the polarity of microtubule-based motors by fluorescence microscopy. *Methods in cell biology* **39**, 105-113 (1993).

99 Aggarwal, T., Materassi, D., Davison, R., Hays, T. & Salapaka, M. Detection of Steps in Single Molecule Data. *Cellular and Molecular Bioengineering* **5**, 14-31 (2012).

100 Dumont, E. L. P. & Hess, H. Observing the mushroom-to-brush transition for kinesin proteins. *Langmuir, Submitted* (2013).

101 Toprak, E., Yildiz, A., Hoffman, M. T., Rosenfeld, S. S. & Selvin, P. R. Why kinesin is so processive. *P Natl Acad Sci USA* **106**, 12717-12722 (2009).

102 VanBuren, V., Cassimeris, L. & Odde, D. J. Mechanochemical model of microtubule structure and self-assembly kinetics. *Biophys J* **89**, 2911-2926 (2005).

103 Agarwal, A., Katira, P. & Hess, H. Millisecond curing time of a molecular adhesive causes velocity-dependent cargo-loading of molecular shuttles. *Nano Lett.* **9**, 1170-1175 (2009).

104 Neuert, G., Albrecht, C. H. & Gaub, H. E. Predicting the rupture probabilities of molecular bonds in series. *Biophys J* **93**, 1215-1223 (2007).

105 Bhaskaran, H. *et al.* Ultralow nanoscale wear through atom-by-atom attrition in silicon-containing diamond-like carbon. *Nat Nanotechnol* **5**, 181-185 (2010).

106 Gnecco, E., Bennewitz, R. & Meyer, E. Abrasive wear on the atomic scale. *Phys Rev Lett* **88** (2002).

107 Gotsmann, B. & Lantz, M. A. Atomistic wear in a single asperity sliding contact. *Phys Rev Lett* **101** (2008).

108 Howard, J., Hunt, A. & Baek, S. Assay of Microtubule Movement Driven by Single Kinesin Molecules. *Method Cell Biol* **39**, 137-147 (1993).

109 Yurke, B., Turberfield, A. J., Mills, A. P., Simmel, F. C. & Neumann, J. L. A DNA-fuelled molecular machine made of DNA. *Nature* **406**, 605-608 (2000).

110 Soong, R. K. *et al.* Powering an inorganic nanodevice with a biomolecular motor. *Science* **290**, 1555-1558. (2000).

111 Hess, H., Bachand, G. D. & Vogel, V. Powering Nanodevices with Biomolecular Motors. *Chem. Eur. J.* **10**, 2110-2116 (2004).

112 Howard, J. *Mechanics of Motor Proteins and the Cytoskeleton*. (Sinauer, 2001).

113 Brunner, C., Hess, H., Ernst, K.-H. & Vogel, V. Lifetime of biomolecules in hybrid nanodevices. *Nanotechnology* **15**, S540-S548 (2004).

114 Katira, P. *et al.* Quantifying the performance of protein-resisting surfaces at ultra-low protein coverages using kinesin motor proteins as probes. *Advanced Materials* **19**, 3171-3176 (2007).

115 Tucker, R. *et al.* Temperature Compensation for Hybrid Devices: Kinesin's K-m is Temperature Independent. *Small* **5**, 1279-1282 (2009).

116 Vigers, G. P. A., Coue, M. & McIntosh, J. R. Fluorescent Microtubules Break Up Under Illumination. *J. Cell Biol.* **107**, 1011-1024 (1988).

117 Vale, R. D., Schnapp, B. J., Reese, T. S. & Sheetz, M. P. Organelle, bead, and microtubule translocations promoted by soluble factors from the squid giant axon. *Cell* **40**, 559-569 (1985).

118 Leduc, C., Ruhnow, F., Howard, J. & Diez, S. Detection of fractional steps in cargo movement by the collective operation of kinesin-1 motors. *PNAS* **104**, 10847-10852 (2007).

119 Wettermark, G., Borglund, E. & Brolin, S. E. A regenerating system for studies of phosphoryl transfer from ATP. *Anal. Biochem.* **22**, 211-218. (1968).

120 Wilson, C. A. *et al.* Myosin II contributes to cell-scale actin network treadmilling through network disassembly. *Nature* **465**, 373-U137 (2010).

121 Evans, E. Probing the relation between force--lifetime--and chemistry in single molecular bonds. *Annu. Rev. Biophys. Biomol. Struct.* **30**, 105-128. (2001).

122 Goodsell, D. S. Biomolecules and Nanotechnology. *Am. Sci.* **88**, 230 (2000).

123 Asbury, C. L., Fehr, A. N. & Block, S. M. Kinesin Moves by an Asymmetric Hand-Over-Hand Mechanism. *Science* **302**, 2130-2134 (2003).

124 Kerssemakers, J. W. J. *et al.* Assembly dynamics of microtubules at molecular resolution. *Nature* **442**, 709-712 (2006).

125 Lee, G., Hartung, S., Hopfner, K. P. & Ha, T. Reversible and Controllable Nanolocomotion of an RNA-Processing Machinery. *Nano Lett.* **10**, 5123-5130 (2010).

126 Sweeney, H. L. & Houdusse, A. Structural and Functional Insights into the Myosin Motor Mechanism. *Annual Review of Biophysics* **39**, 539-557 (2010).

127 Bissell, R. A., Cordova, E., Kaifer, A. E. & Stoddart, J. F. A Chemically and Electrochemically Switchable Molecular Shuttle. *Nature* **369**, 133-137 (1994).

128 Wickham, S. F. J. *et al.* Direct observation of stepwise movement of a synthetic molecular transporter. *Nat Nano* **6**, 166-169 (2011).

129 Kay, Euan R., Leigh, David A. & Zerbetto, F. Synthetic Molecular Motors and Mechanical Machines. *Angewandte Chemie International Edition* **46**, 72-191 (2007).

130 Merkel, R., Nassoy, P., Leung, A., Ritchie, K. & Evans, E. Energy landscapes of receptor-ligand bonds explored with dynamic force spectroscopy. *Nature* **397**, 50-53. (1999).

131 Zhurkov, S. N. Kinetic Concept of Strength of Solids. *International Journal of Fracture Mechanics* **1**, 311-323 (1965).

132 Willis, M. S., Schisler, J. C., Portbury, A. L. & Patterson, C. Build it up-Tear it down: protein quality control in the cardiac sarcomere. *Cardiovasc. Res.* **81**, 439-448 (2009).

133 Vale, R. D. & Milligan, R. A. The way things move: looking under the hood of molecular motor proteins. *Science* **288**, 88-95 (2000).

134 Liu, Y. *et al.* Linear artificial molecular muscles. *J. Am. Chem. Soc.* **127**, 9745-9759 (2005).

135 Bell, G. I. Models for Specific Adhesion of Cells to Cells. *Science* **200**, 618-627 (1978).

136 Guzman, D. L., Randall, A., Baldi, P. & Guan, Z. B. Computational and single-molecule force studies of a macro domain protein reveal a key molecular determinant for mechanical stability. *Proc. Natl. Acad. Sci. U. S. A.* **107**, 1989-1994 (2010).

137 Wendler, K., Thar, J., Zahn, S. & Kirchner, B. Estimating the Hydrogen Bond Energy. *J Phys Chem A* **114**, 9529-9536 (2010).

138 Mammen, M., Choi, S. K. & Whitesides, G. M. Polyvalent interactions in biological systems: Implications for design and use of multivalent ligands and inhibitors. *Angew Chem Int Edit* **37**, 2755-2794 (1998).

139 Pincet, F. & Husson, J. The solution to the streptavidin-biotin paradox: The influence of history on the strength of single molecular bonds. *Biophys. J.* **89**, 4374-4381 (2005).

140 Howard, J. & Hymann, A. A. Dynamics and mechanics of the microtubules plus end. *Nature* **422**, 753-758 (2003).

141 Margolis, R. L. & Wilson, L. Microtubule treadmilling: what goes around comes around. *Bioessays* **20**, 830-836. (1998).

142 Loparo, J. J., Kulczyk, A. W., Richardson, C. C. & van Oijen, A. M. Simultaneous single-molecule measurements of phage T7 replisome composition and function reveal the mechanism of polymerase exchange. *Proceedings of the National Academy of Sciences* **108**, 3584-3589.

143 Shi, X. & Ha, T. Seeing a molecular machine self-renew. *Proceedings of the National Academy of Sciences*.

144 Veigel, C. & Schmidt, C. F. Friction in Motor Proteins. *Science* **325**, 826-827 (2009).

145 Marden, J. H. & Allen, L. R. Molecules, muscles, and machines: universal performance characteristics of motors. *Proc. Natl. Acad. Sci. U. S. A.* **99**, 4161-4166. (2002).

146 Ashby, M. *Materials Selection in Mechanical Design*. 3 edn, (Butterworth-Heinemann, 2005).

147 Schmiedl, T. & Seifert, U. Efficiency of molecular motors at maximum power. *Epl-Europhys Lett* **83**, 30005 (2008).

148 Kellermayer, M. S. Z., Smith, S. B., Bustamante, C. & Granzier, H. L. Mechanical fatigue in repetitively stretched single molecules of titin. *Biophys. J.* **80**, 852-863 (2001).

149 Roos, W. H., Bruinsma, R. & Wuite, G. J. L. Physical virology. *Nat Phys* **6**, 733-743 (2010).

150 Feder, J. Random Sequential Adsorption. *Journal of Theoretical Biology* **87**, 237-254 (1980).

151 Talbot, J., Tarjus, G., Van Tassel, P. R. & Viot, P. From car parking to protein adsorption: an overview of sequential adsorption processes. *Colloid Surface A* **165**, 287-324 (2000).

152 Manciu, M. & Ruckenstein, E. Estimation of the available surface and the jamming coverage in the Random Sequential Adsorption of a binary mixture of disks. *Colloid Surface A* **232**, 1-10 (2004).

153 Van Tassel, P. R. *et al.* A particle-level model of irreversible protein adsorption with a postadsorption transition. *J Colloid Interf Sci* **207**, 317-323 (1998).

154 Adamczyk, Z., Barbasz, J. & Ciesla, M. Mechanisms of Fibrinogen Adsorption at Solid Substrates. *Langmuir* **27**, 6868-6878 (2011).

155 Weaver, D. R. & Pitt, W. G. Sticking Coefficients of Adsorbing Proteins. *Biomaterials* **13**, 577-584 (1992).

156 Jung, L. S. & Campbell, C. T. Sticking probabilities in adsorption of alkanethiols from liquid ethanol solution onto gold. *J Phys Chem B* **104**, 11168-11178 (2000).

157 Ramm, L. E., Whitlow, M. B. & Mayer, M. M. The Relationship between Channel Size and the Number of C9 Molecules in the C5b-9 Complex. *J Immunol* **134**, 2594-2599 (1985).

APPENDICE

Publications and Presentations

Life is an absurd wandering toward a certain death

Stig Dagerman, Our need of consolation is impossible to satisfy

Peer-reviewed publications

- H. Hess and **E.L.P. Dumont**: “Fatigue Failure and Molecular Machine Design”, Small, 2011, 7, 1619-1623.
- **E.L.P. Dumont**, H. Belmas, and H. Hess: “Observing the mushroom-to-brush transition for kinesin proteins, Langmuir, 2013, 29 (49), 15142-15145.
- **E.L.P. Dumont** and H. Hess: “Molecular wear of microtubules propelled by surface-adhered kinesins”, Nature Nanotechnology, under review.
- Y. Jeune-Smith, **E.L.P. Dumont,** and H. Hess: “Wear and breakage combine to mechanically degrade kinesin-powered molecular shuttles”, in preparation.
- **E.L.P. Dumont,** A. Guillaume, and H. Hess: “Random Sequential Adsorption of proteins on polymer-covered surfaces: A simulation-based approach”, in preparation.
- **E.L.P. Dumont** and H. Hess: “A molecular model to explain material fracture at low stresses”, in preparation.

Oral presentations

- **Complexity and Intuition**, TEDx Columbia Engineering, 2012.
- **Fatigue, friction, and wear at the molecular scale**, The US Turkey Advanced Study Institute on Global Healthcare Grand Challenges, 2012.

Poster presentations

- Y. Jeune-Smith, **E.L.P. Dumont**, K. Nittala and H. Hess: "Protein Friction Causes Molecular Wear in Active Nanosystems", 2010 ICMR Summer School on Nanoscale Science of Biological Interfaces, Santa Barbara, CA (6/19/2010-7/1/2010).

- P. Katira, A. Agarwal, Y. Jeune-Smith, **E.L.P. Dumont**, K. Nittala, and H. Hess: "Proteins at Interfaces: Non-fouling Surfaces and Nanotribology", Gordon Research Conferences Biointerface Science, Les Diablerets, Switzerland (9/5/2010-9/10/2010).
- I.K. Jun, **E.L.P. Dumont**, and Henry Hess: "A biomimetic, self-pumping membrane", BMES Annual Meeting, Hartford, CT (10/2011).
- **E.L.P. Dumont** and H. Hess: "Fatigue, friction, and wear at the molecular scale", 56th Annual meeting of the Biophysical Society, San Diego, CA (2/25/2012-2/29/12).
- H. Hess, **E.L.P. Dumont**, Y. Jeune-Smith, K. Nittala: "Accelerated Degradation of Active Nanosystems by Biomolecular Motors", NSF CMMI Grantees Conference, Boston, (7/9/2012-7/12/2012).
- **E.L.P. Dumont** and H. Hess: "Fatigue, friction, and wear at the molecular scale", The US Turkey Advanced Study Institute on Global Healthcare Grand Challenges, Antalya, Turkey, (7/13/12).
- **E.L.P. Dumont** and H. Hess: "Protein Friction Causes Molecular Wear in Kinesin-powered Molecular Shuttles", 57th annual meeting of the Biophysical society, Philadephia (2/15/13).
- **E.L.P. Dumont** and H. Hess: "Proteins at interfaces: Conformational behavior and wear", Peptides and Proteins, Le Mans, France (7/30/13).