2-8-2011

Kinetic and Statistical Mechanical Modeling of DNA Unzipping and Kinesin Mechanochemistry

Lawrence Herskowitz

Lawrence J. Herskowitz

Candidate

Physics and Astronomy

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

,Chairperson

KINETIC AND STATISTICAL MECHANICAL MODELING OF DNA UNZIPPING AND KINESIN MECHANOCHEMISTRY

BY

LAWRENCE J. HERSKOWITZ

B.S., Physics, State University of New York at Albany, 2006
B.S., Mathematics, State University of New York at Albany, 2006

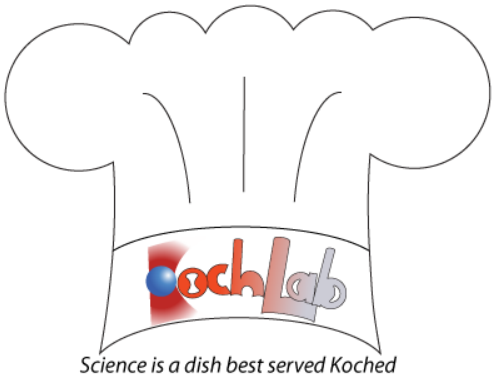
DISSERTATION

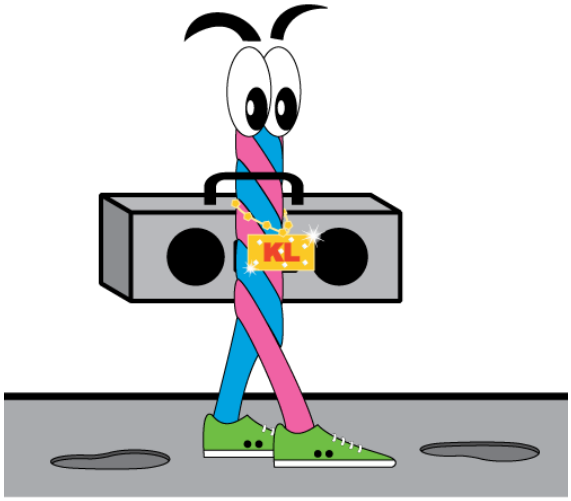
Submitted in Partial Fulfillment of the
Requirements for the Degree of

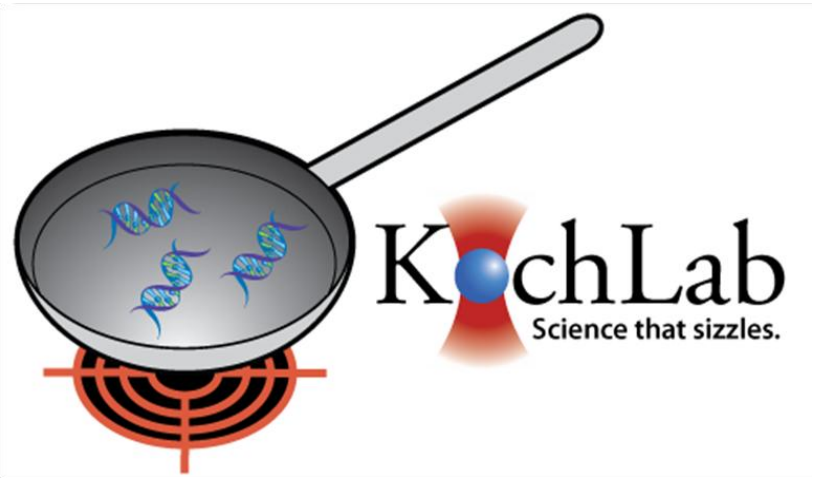
Doctor of Philosophy
Physics

The University of New Mexico
Albuquerque, New Mexico

December, 2010


KochLab
Science is a dish best served Koched


KL


KochLab
Science that sizzles.

Acknowledgements

There are a lot of people without who I would not be graduating. I guess I'll thank them chronologically when they entered my life and helped me along the way.

I have to thank my parents, Joni and Wayne Herskowitz. Obviously without them I wouldn't be here today. They gave me a good life and took care for me before I knew how. They sacrificed a lot for me, and I thank them for that. They instilled their work ethic in me. Without that I wouldn't have been able to finish this dissertation so quickly.

Thanks to my brothers: Mike Herskowitz for giving me someone to talk to about comics (seriously there is no one around here who wants to talk about comics) and Eric Herskowitz for looking up to me and thus giving me another reason to be a better person.

Next I have to thank Mrs. Traci Maier, my 11th grade physics teacher. Back when I was just a little Larry I was heavily interested in biology, but in my high school the AP course was offered every other year. So to kill time before I could take AP biology I took AP physics with Mrs. Maier. She was the first person to introduce me to vectors, gravity, two dimensional projectiles, all that first year stuff. But most importantly since she was such a fantastic teacher she sparked my love of physics. In fact she was so good that I took a higher level AP physics course the next year instead of biology. However the teacher I got the next year was horrific, but my love for physics was too strong and he couldn't kill it. If it wasn't for Mrs. Maier I would not have a Ph.D. in physics today.

After high school I went to SUNY Albany and met Anthony Salvagno. He was also a physics major, and we became very close friends. We did more than keep each other out of trouble. We made each other better students. We mad each other better athletes. We made each other better. We shared a lot of experiences together, and we have tons of stories. He was there for my highs and lows, and if it wasn't for him I wouldn't have been able to make it this far.

At the University of New Mexico I would like to thank my friends who made the stress of grad school a bit easier. In particular I'd like to acknowledge Brad Chase and Dara Easley. I thank Brad Chase for his near perfection in all aspects of life. Thanks to Dara Easley for her love to cook delicious cookies but not wanting to eat them. In fact I'd like to thank her twice for that.

In their roles as teachers, mentors and advisors, I'd like to thank the following: Dr. Plotnick, Dr. Caticha, Dr. Ernst, Dr. Childs from SUNY Albany, and Dr. Prasad, Mickey Odom, Dr. Deutsch, Dr. Dunlap, Dr. Evans from UNM.

Thanks to everyone from Koch Lab. Anthony Salvagno who I gave a paragraph to above so it would be redundant to write more here. Hey! How much space do you want in the acknowledgements section? Andy Maloney for his quirky behavior which I have never made fun of and his love of doughnuts. Also I have to thank him for finding each and every bug in all of the programs I wrote for him. That has made me a better programmer but dang was that annoying. Pranav Rathi and Nadia Oropeza you guys came late to the lab and I didn't work with either of you that much, but thanks for being such a great second generation

Kochers. Keep the lab strong when I'm gone. Also like to thank Igor Kuznetsov for all the help, but not for giving me food poisoning that once. That kind of stunk.

Of course I'd be remised if I didn't thank my advisor Dr. Steven Koch. Dr. Koch is a fantastic advisor, mentor, and boss. He is one of the smartest people I know, and also a great teacher. He has a lot of patience. And he needed it to teach me. So thanks for the patience to see through all of our arguments. Thanks also from transcending boss to friend.

To Ginny Morriss, thanks for your strength, compassion and support. Thanks for your encouragement especially when I needed to work 70 hours a week to get my work done. A special thanks for listening to me prattle on about my work, Colorado Rockies, and comic books.

KINETIC AND STATISTICAL MECHANICAL MODELING OF DNA UNZIPPING AND KINESIN MECHANOCHEMISTRY

BY

LAWRENCE J. HERSKOWITZ

ABSTRACT OF DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Physics

The University of New Mexico
Albuquerque, New Mexico

December, 2010

Kinetic and Statistical Mechanical Modeling of DNA Unzipping and Kinesin Mechanochemistry

Lawrence J. Herskowitz

B.S., Physics, State University of New York at Albany
B.S., Mathematics, State University of New York at Albany
Ph.D., Physics, University of New Mexico

ABSTRACT

This thesis explores two topics. The first is shotgun DNA mapping (SDM). Ability to map polymerases and nucleosomes on chromatin is important for understanding the impact of chromatin remodeling on key cellular processes. Current methods have produced a wealth of information that demonstrates this importance, but key information is elusive in these methods. We are pursuing a new single-molecule chromatin mapping method based on unzipping native chromatin molecules with optical tweezers. The first step we are taking towards this ability is SDM. This is the ability to identify the genomic location of a random DNA fragment based on its naked DNA unzipping forces compared with simulated unzipping forces of a published genome. We show that ~32 separate experimental unzipping curves for pBR322 were correctly matched to their simulated unzipping curves hidden in a background of the ~2700 sequences neighboring XhoI sites in the *S. cerevisiae* (yeast) genome. We describe this method and characterize its robustness as well as discuss future applications.

The second topic is a discrete state model for kinesin-1's processivity. Kinesin-1 is a homodimeric molecular motor protein that uses ATP and a hand-over-hand motion to transport cargo along microtubules. Minimal kinetic models are

often developed to both explain kinesin's hand-over-hand forward-stepping behavior and to infer important kinetic rate constants from experimental data. These minimal models are often limited to a handful of two-headed states on a core cycle. However, it is not always clear how to evolve these core-cycle models to explain more complex behavior. We have developed a kinetic model without a pre-defined core cycle. Our model includes 80 two-headed states and permits transitions between any two states that differ by a single catalytic or binding event. We constrain the rate constants as much as possible by published rates and mechanical strain in the kinesin neck linkers and their docking state. We present a model for neck-linker modulation of head and nucleotide binding and unbinding rates. We show that our model reproduces generally-accepted experimental results. The core cycles that emerge are slightly different than those seen in previous experiments. We also explore how processivity and speed change with neck linker length.

Table of Contents

List of Figures

List of Tables

Introduction

The majority of this dissertation is devoted to using statistical mechanics and kinetics to study two distinct problems. The first is the development of a novel mapping technique which uses a statistical mechanical model to predict the forces it takes to break double stranded DNA bonds. The second is the application of a discrete state model to probe how a motor protein, kinesin, can walk. The core of that model uses statistical mechanics to calculate rate constants. This dissertation is divided into four research-orientated chapters. All of these chapters except Chapter 4 is a reprint of a previously or soon-to-be-published work.

Chapter 1 describes in detail the process of Shotgun DNA Mapping (SDM). SDM is a term we coined that refers to the possible ability to identify short DNA strands by comparing its unzipping force curves to simulated force curves. DNA in eukaryotes exists in the form of chromatin. In this structure DNA is tightly packed and wrapped around histones to form a nucleosome (Kornberg 1977). The nucleosomes act as barriers and as a regulation mechanism during DNA replication, gene transcription, and DNA repair (Workman and Kingston 1998; Khorasanizadeh 2004). Nucleosome remodeling is the process of moving or completely removing the nucleosome. Chromatin Immunoprecipitation (ChIP) is a tool currently used to study this process (Sudarsanam 2000; Martens 2003). While ChIP has provided great insight, improvement in the spatial and temporal resolution is necessary to further study nucleosome remodeling. Utilizing a single-molecule technique such as SDM will provide these needed improvements.

Using an optical tweezer, it has been shown that nucleosomes, RNA polymerases, and other proteins attached to the double stranded DNA (dsDNA) can be located with high spatial resolution (Hall *et al*. 2009; Shundrovsky *et al*. 2006). This is achieved by unzipping the DNA usually with an optical tweezer (OT) and "popping off" any bound proteins. This causes a spike in the force required to unzip the DNA. After all proteins have been removed, the experimenter can decreases the force applied by the OT, allowing the unzipped single stranded DNA (ssDNA) to rezip into dsDNA. Unzipping the DNA again will give the unzipping curves of the now naked DNA. Since the curve is sequence dependent, it can be compared to a library of unzipping curves simulated by the software explained in Chapter 1 to map the locations of the bound proteins onto the genome.

The bulk of the work discussed herein focuses on the study of kinesin-1 (called kinesin for simplicity). Kinesin is a motor protein found in neurons. Studies have shown kinesin is essential for proper neural activities (Gindhart 2006). In humans neurons can be up to a meter in length and can grow even longer in larger animals. Molecules necessary for neuronal mechanisms including saltatory movement are produced in the soma but are needed in the dendrites or axon. Even within the axon molecules are needed in specific locations, for example an ankyrin isoform is needed in the nodes of Ranvier and synaptic proteins are required in the axon terminal (De Vos *et al*. 2008; Kordeli *et al*. 1990). Over small distances diffusion can theoretically account for molecular transportation. However if the neuron relied solely on diffusion for transportation of its vital molecules, it could take approximately 10^{10} seconds to reach their destinations.

Neurons (and many other cells types) have motor proteins to actively transport important molecules by using a scaffolding system of polymers including microtubules. Motor protein transportation in neurons can be classified as retrograde or anterograde. Retrograde transport means cargo is being brought to the soma. Anterograde transport means the molecule is being taken away from the soma. In neurons, transport can also be classically categorized as fast or slow axonal transport, defined by a large measured difference in the speed of different molecules being transported. It is now known that the slower speed is due to long pauses rather than different molecular motors transporting the cargo (Roy *et al*. 2000; L Wang *et al*. 2000). Since our focus here is on motors rather than cargo, it is apt to elucidate the directionality classification system since it separates the process largely according to motor type. Dynein is chiefly responsible for retrograde transport while kinesin has a predominant role in the anterograde transport of molecules (Hirokawa and Takemura 2005; Hirokawa 1998; Hirokawa *et al*. 1991; Goldstein and Yang 2000; Schnapp and Reese 1989).

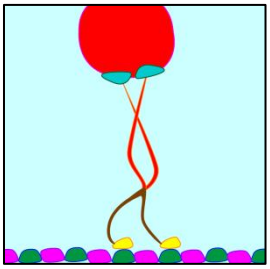


Figure 1.1. Schematic representation of kinesin. This illustration is color coded for easier identification. Examining this illustration from top to bottom, we see the cargo symbolized by a large red circle binds to the two teal tails. In reddish orange is the stalk. The neck linker, colored brown, connects the two heads that are colored yellow. The microtubule is comprised of alternating alpha (magenta) and beta (green) tubulin dimers. The kinesin heads bind to the to the beta tubulin.

Kinesin, seen in Figure 1.1, is comprised of two heads (seen in yellow) which anchor the protein to the microtubule, a neck linker connecting the two heads (brown), a stalk (reddish orange) and tails (teal). The cargo (red circle) binds with

the tail through protein-protein interactions and with some help from a population of regulatory proteins. Kinesin transports molecules by “walking” along a microtubule. A microtubule is a polymer consisting of consecutive beta and alpha tubulin dimers (seen in Figure 1.1 as green and magenta respectively). These polymers act as tracks to guide kinesin’s stepping process. The kinesin heads can bind to the beta tubulin and through a series of stochastic transitions including the hydrolysis of ATP, the back kinesin head will unbind from the beta tubulin. It then steps in front of the other head and binds to the next beta tubulin in front of the other head. The cycle then starts again but with the other head detaching from the microtubule. It is important to note that, even though kinesin is said to walk and it has heads despite this personification kinesin is not a sentient being. Walking can only occur by cycling through a series of states where it is more probable for the kinesin to step forward instead of backwards or detaching. This process is the basis of the research described in Chapter 2.

Defects in this active transport mechanism can lead to many diseases. Model systems such as *Drosophila* (*Sophophora* (O’Grady 2010; O’Grady and Markow 2009)) *melanogaster* or *Caenorhabditis elegans* are used to study mutations in kinesin to better understand some neurodegenerative diseases (Goldstein and Yang 2000; Morfini *et al*. 2009; Muqit and Feany 2002; Wu and Luo 2005; Hurd and Saxton 1996). For example it has been shown that a kinesin mutation impairing kinesin function in *Drosophila*, leads to neuronal swelling in the axons with accumulations of transported vesicles, synaptic membranes and mitochondria which kinesin typically transports. This defect resulted in loss of muscle control

function and larval lethality (Hurd and Saxton 1996). Similar swelling has been seen in many neurodegenerative diseases for example amyotrophic lateral sclerosis (ALS or Lou Gehrig's Disease) and Alzheimer's Disease (Coleman 2005; De Vos *et al*. 2008; Stokin *et al*. 2005). It is not clear what is exactly disturbing axonal transport in some of these cases. Factors which may inhibit axonal transport include disrupting the microtubule, inhibiting the binding of cargo, mutations in the motor protein, or loss of mitochondrial function and decreasing ATP concentration. Hereditary Spastic Paraplegia Type 10 (HSP(SPG10)) is an example of a disease caused by a mutation in kinesin heavy chain genes, resulting in the inhibition of microtubule-bound ATPase activity

Our lab has two ways to study kinesin. Chapter 2 describes the use of a stochastic model of kinesin's stepping process using discrete states. This chapter also describes analysis of the model using a state machine and Markov Chain theory. We also study kinesin's velocity through a gliding motility assay, one of the first assays used to study kinesin (Vale, Reese, and Sheetz 1985). In this assay kinesin is affixed to a cover slide with the motor domains away from the cover glass while microtubules are allowed to move on their heads (Figure 1.2). Visualization using a microscope, allows the microtubule movement to be viewed, as shown in the false colored image in Figure 1.3.

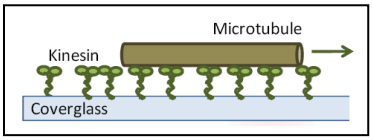


Figure 1.2. Schematic illustration of gliding motility assay. The kinesin is attached to the cover glass with its heads are exposed to the solution. This immobilizes the kinesin but still allows it to walk along microtubules. When this happens the microtubule is seen as moving.

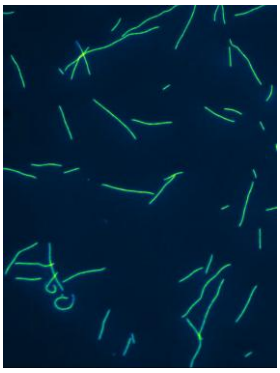


Figure 1.3. False colored image from a gliding motility assay. The microtubules are false colored green in this image. This assay was performed by Andy Maloney (Maloney, Herskowitz, and Koch 2011).

To study the velocity of kinesin microtubules in each experiment need to be tracked. We have created custom software to track microtubules and then using the polymer's trajectory and time between frames we calculate the microtubule's speed. We created software that can mimic the images seen in a gliding motility assay to test the effectiveness of our tracking software. This image simulation software and the tracking software are explained in Chapters 3 and 4 respectively.

This dissertation is divided into two parts. Chapter 1 describes SDM and Chapters 2-4 describe the kinesin work.

Proof of Principle for Shotgun DNA Mapping by Unzipping

Lawrence J. Herskowitz, Anthony L. Salvagno, R. Andy Maloney, Linh N. Le and Steven J. Koch

Introduction

Chromatin remodeling affects the ability of other proteins to access the DNA and has a profound impact on fundamental processes such as DNA repair and gene transcription by RNA polymerase. Understanding of these dynamic remodeling processes requires the ability to characterize with high spatial and temporal resolution the changes to chromatin inside living cells. Techniques such as chromatin Immunoprecipitation (ChIP), ChIP-chip, and other existing techniques have provided a wealth of important information, but have drawbacks in terms of sensitivity to small changes in protein occupancy, spatial resolution, and ensemble averaging. Certain information can only be obtained via single-molecule (SM) analysis, such as seeing direct correlations between polymerases and nucleosomes on individual fibers or differentiating between some proposed models of chromatin remodeling (Boeger, Griesenbeck, and Kornberg 2008).

To obtain this type of information, we are developing a single-molecule method for mapping polymerases and nucleosomes on chromatin based on optical tweezers unzipping of native chromatin molecules. It has been shown that SM DNA unzipping can map the positions of mononucleosomes assembled *in vitro* based on a repeatable nucleosome unzipping force profile (Shundrovsky *et al*. 2006). We expect RNA Polymerase II (Pol II) complexes to also have a repeatable unzipping force profile, but distinguishable from nucleosomes and perhaps also indicating the sense / antisense orientation of the Pol II. The Pol II data is not yet available, but if it is as expected, then we anticipate that SM unzipping of

native chromatin fragments (extracted from living cells) will provide high-resolution mapping of nucleosomes and Pol II molecules (along with orientation) on individual chromatin fibers.

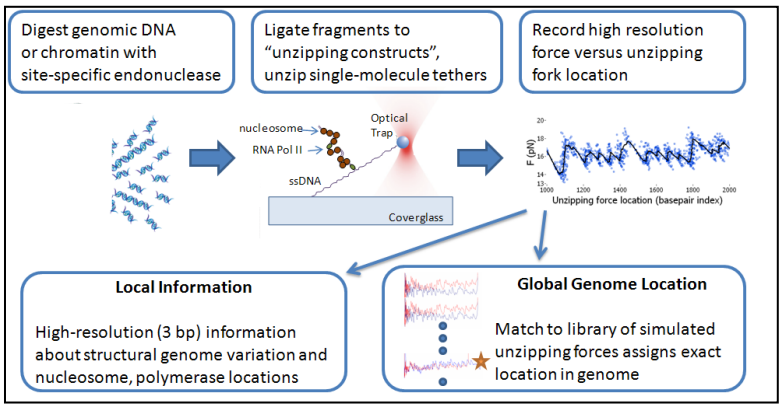


Figure 2.2. Overview of proposed method for shotgun DNA and chromatin mapping. We have recently achieved proof-of-principle results important for the "Global Genome Location," part of the process (*lower right*).

We may be able to obtain important information from high-resolution SM mapping on individual fragments, even if their specific location in the genome were unknown. For example, the electron microscopy analysis of chromatin and RNA transcripts has demonstrated the utility of SM information even when the identity of the genes was unknown (Sikes, Beyer, and Osheim 2002). However, it would be much more powerful and thus desirable to obtain high-resolution SM information about specific genes or other sites in the genome of interest. For example, site-specific SM analysis may provide crucial insight into the issues of promoter-proximal Pol II pausing and antisense transcription which have recently been shown to be very important (Margaritis and Holstege 2008; He *et al*. 2008; Core, Waterfall, and Lis 2008; Muse *et al*. 2007; Zeitlinger *et al*. 2007; Buratowski 2008; Core and Lis 2008). Thus, we are pursuing methods for *site-specific* SM analysis of chromatin. The first way we have tried to do this is by engineering unique restriction sites into the yeast genome (I-SceI) at a specific site. This has

proven difficult, and has the disadvantage of requiring genetic engineering of all mutant strains and cell types that will be analyzed. Thus, we are now pursuing a second way of achieving site-specificity which is to unzip random chromatin fragments in a high-throughput fashion, and then figuring out from which specific site of the genome it came. We call this shotgun chromatin mapping (SCM) and it based on a method for indentifying the genomic location of naked DNA fragments (see Fig. 2.1).

It has been shown that the unzipping forces for a known sequence of DNA can be accurately predicted by statistical mechanical models (Koch *et al*. 2002; Bockelmann *et al*. 2002). Furthermore, at this time many genomes have been published and the number is rapidly increasing. These two facts together led us to believe that the naked DNA experimental unzipping forces would allow us to identify the genomic location of random DNA fragments. We call this process shotgun DNA mapping (SDM). The basic procedure is to compare an unknown fragment's force data to a library of known possible fragments' simulated unzipped force data. The fragment possibilities can be limited, for example, by digestion with a site-specific restriction endonuclease. In a successful method, the experimental data will reliably match up the best with the simulation of its true sequence. The identify of a DNA fragment could be easily identified manually ("by eye") from among a handful of possibilities, but it remains to be shown whether the simulations can be accurate enough for automated identification a fragment from the background of thousands of fragments expected from site-specific digestion of genomic DNA.

In this paper, we show that SDM is possible. Specifically, we demonstrate that the modeling of the pBR322 unzipping forces is sufficiently accurate so that experimental data are successfully matched to the pBR322 sequence hidden in a background of the ~2700 XhoI fragments from the yeast genome. We explain our methods, show where to obtain our software and data, and discuss further potential improvements which indicate it will also be successful with much larger fragment libraries. We feel this technique will be a key enabler of our goal of shotgun chromatin mapping. Furthermore, we envision other high impact applications, for example single-molecule structural genome mapping (Kidd *et al*. 2008) and new assays for screening protein binding sites by shotgun DNA mapping in the presence of purified proteins.

Methods

All computations below were carried out using a Dell duoCore running Windows XP. Code was written in LabVIEW 7.1

Experimental Single-Molecule Unzipping Data

We obtained force (F) versus unzipping index (j) for 32 data sets of unzipping pBR322 from the published data of Koch *et al*. (Koch *et al*. 2002). Data were obtained and analyzed with optical tweezers and unzipping constructs as described. The format of these data sets is tab delimited text files, with the “Force (pN)” and “index (j)” columns used by us. The 32 raw data sets are available on http://kochlab.org . We used particular data sets which seem to have significant viscous drag due to high stretching rate.

Data were smoothed according to a sliding boxcar smoothing algorithm we implemented in LabVIEW. We used a 30 point window with equal weighting to each point in the window, and a window step size of j=1. Smoothed data sets were stored in text files of the same format as the simulated data (below) and will be available on http://kochlab.org or upon request.

Extraction of Yeast Genome XhoI Sites

We obtained the yeast genome (S. Cerevisiae) from yeastgenome.org. We downloaded a text file for each chromosome of the yeast genome. We wrote a LabVIEW application to do the following:

- Read in a sequence text file
- Eliminate white space and non-DNA base information
- Search for XhoI recognition sites (CTCGAG)
- For each recognition site two fragments were formed, 2000 base pairs before the site and 2000 base pairs after. These were stored as text files sequence only with a naming convention Chromosome Number/Recognition Site Index, Downstream or Upstream. These files will be available on http://kochlab.org. Upstream fragments are reversed so as to begin with the XhoI recognition site.
- Additionally the pBR322 sequence used in Koch 2002 was manually added to the sequence library with a code name to blind it from the data analyzers.
    - The pBR322 fragment used for the experimental work was created from EarI digestion of the plasmid.

Creation of Simulation Library for Yeast XhoI Sites

Methods for simulating unzipping of the above fragments was as in previous work by Bockelmann *et al*. with slight modifications (Bockelmann *et al*. 2002). Our Hamiltonian was:

$$E_{total} = E_{DNA} + E_{FJC} \tag{2.1}$$

where E_{DNA} accounts for the energy of the base pairing, and E_{FJC} is the mechanical energy of the stretched ssDNA. This is simplified from the work of Bockelmann *et al*. (Bockelmann *et al*. 2002) by not including the optical tweezer energy. Further we ignored elastic energy from the dsDNA anchoring fragment used in the unzipping experiments (Koch *et al*. 2002). E_{DNA} for a given j is the sum of E_i where E_i is (1.3 k_BT for A-T or 2.9 k_BT for G-C). Values for E_i were obtained from Bockelmann et al. (Bockelmann *et al*. 2002). E_{FJC} for a given j and l is

$$E_{FJC} = xF - \int_0^F x(F')dF' \tag{2.2}$$

x(F) is the extensible FJC (MD Wang *et al*. 1997). We used FJC values from Koch 2002 (Koch *et al*. 2002). The integral was computed numerically. The F for a given x was found using Newton's Method.

We wrote an algorithm in LabVIEW to calculate expectation values for F, j, and the variance of each for a given DNA sequence and end to end length, l. The expectation values were calculated by simple sums over all possible j values (from 1 to the length of the sequence). Simulated F versus j curves were then generated by embedding this algorithm in a loop that stepped over varying values for l. An automated process loaded each sequence and produced F versus j curves for all yeast XhoI fragments in the library. For this work, the expectation

values were calculated in steps of 1 nm from 1 nm to 2200 nm and sums over j from 1 to 2000. Simulation results were stored in text files, one file for each XhoI fragment and will be available from http://kochlab.org.

Matching Algorithms

We devised an algorithm that can produce a quantitative measure of the similarity between two force versus j curves. We call this measure the match score (m), and it is derived from the standard deviation of the two curves in a given interval. To compute m we used this formula:

$$m = \frac{k_B T}{2C_0} \frac{N}{\sqrt{\sum_i^N (< F_i^{\exp} > - < F_i^{sim} >)^2}} \tag{2.3}$$

where F^{exp} and F^{sim} are the experimental and simulated unzipping forces respectively (as a function of j), k_B is the Boltzman constant, T is temperature, and C_0 is the single-stranded DNA contour length per nucleotide.

We wrote a LabVIEW application to calculate the match scores for each experimental data sets against the entire simulation library. These results were stored in a LabVIEW array with each row being one experimental data set. For all match scores in this paper we summed from j = 1200 to j = 1700 in steps of 1. Choice of this window size and location is discussed below and was assisted by a repeatedly running this matching algorithm for various window sizes and locations.

Robustness Analysis

We created a histogram of all incorrect match scores (noise). Then the histogram was fit to a Gaussian using OriginPro (OriginLab Corporation, Northhamp-

ton, MA). A second histogram for all correct match scores was created, and also fit to a Guassian using the same algorithm. An estimate of the robustness was produced by comparing the difference of the means of signal to noise relative to the standard deviation of the noise.

Results

Experimental Single-Molecule Unzipping Data

We smoothed 32 data sets for unzipping of an EarI fragment of pBR322. Examination of force versus unzipping index shows a noticeable increase in the unzipping force for j> 1000. This is due to a significant increase in the unzipping rate above j=1000, because the original purpose of these data sets (Koch 2002) was to probe protein occupancy, where an increased unzipping rate is desirable and a systematic shift in unzipping force is not an issue (Koch *et al*. 2002).

Extraction of Yeast Genome XhoI Sites

We found ~1350 XhoI sites in the yeast genome, which produced a library of ~2700 upstream and downstream unzipping fragments. The entire search and extraction took only a few minutes on our platform. <10 XhoI sites were within 2000 bp from the end of the chromosome, producing fragments less than the desired 2000 bp. These fragments produced nonsense match scores, which were then discarded. Also, by chance, some XhoI sites were separated by less than 2000 base pairs, and thus some fragments included XhoI recognition sequences. In an actual shotgun DNA mapping experiment, these XhoI sites could produce shortened fragments, depending on the level of completion of digestion. We did

not account for this effect in this paper. The resulting library (will be available on http://kochlab.org) included the hidden pBR322 fragment.

Creation of Simulation Library for Yeast XhoI Sites

The force (f) versus unzipping index (j) was simulated for every fragment in the sequence library from l = 1 nm to 2200 nm. Simulation of ~2700 files took approximately a few days on our computational platform. Examples of these simulated curves can be seen in Fig. 2.2A and Fig. 2.2B. Simulations were stored in a library of tab delimited text files.

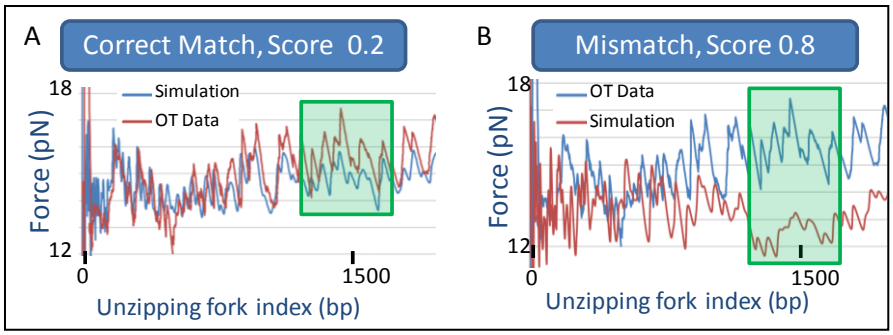


Figure 2.2 Experimental unzipping data compared with (A) correct and (B) incorrect simulation. The green window indicates the region from j=1200 to 1700 where the match scores were computed. The greatly increased separation of the two curves in the incorrect match is reflected in the higher match score of 0.8 versus 0.2 for the correct match.

Matching Algorithms

A key feature of the shotgun DNA mapping process is a mechanism for producing a quantitative number comparing an experimental data set and an entry in the simulation library. We first attempted a cross-correlation algorithm (as in Shundrovsky 2006 (Shundrovsky *et al*. 2006), data not shown), which was unsatisfactory due to the insensitivity of cross-correlation to vertical shifts. That is, the cross-correlation score does not change if the simulation forces are scaled by a factor of 10, for example. Because the unzipping forces reflect the energy of the DNA base pairing, which is directly related to the DNA sequence, absolute un-

zipping force is an important factor in identifying an unknown fragment. Thus, we developed a method based on the standard deviation between the two curves, as described in the methods.

Window Size

The green box highlights the window over which the match scores were computed (j=1200 to 1700). There were a number of reasons for choosing this window size and location. For some shotgun DNA mapping applications, it will be desirable to have the matching window as close to the initial unzipping sequence as possible. However our current implementation of the DNA unzipping simulation does not account for the optical tweezers compliance, or the compliance of the 1.1 kilobases of dsDNA that was used to anchor the segment to the coverglass. This added compliance is critical in the initial unzipping region, where the length of single-stranded DNA is relatively low and thus much stiffer. Thus, we are not yet capable of using this region for our attempts. Furthermore, the data sets we are using have a discontinuous unzipping rate, switching at j ≈1000 from a slow unzipping rate with large data averaging to a fast unzipping rate with no data averaging. Thus, our window must lie on either side of this transition. Neither side is ideal (too much averaging for j<1000 and viscous drag for j>1000), which may demonstrate the robustness of our method. We chose j>1000 due because the large amount of averaging of the raw data during acquisition made the j<1000 data too unappealing.

The ability to use a smaller window size is also desirable for shotgun mapping applications. We investigated the results of smaller window sizes and found that

smaller windows (for example 100 base pairs wide) produced results that were more dependent on the overall location of the window (results from poor to just as good as we show here, data not shown). In contrast, the 500 base pair window was relatively insensitive to location. We chose to use the 500 base pair window so that window location would not significantly affect our proof-of-principle results.

Shotgun Mapping Results

Fig. 2.2A and Fig 2.2B show a comparison of the F versus j curves for the correct match as well as an incorrect match, respectively. By eye, it can easily be seen that there is a larger deviation between the two curves in the Fig 2.2B. This is reflected by the increased white space between the curves, and is effectively what our matching algorithm quantifies, with a score of zero reflecting a perfect match. For this particular data set, the match score was 0.2, and the mismatch shown produced a score of 0.8.

The match scores for this experimental curve against the entire library are shown in Fig. 2.3. In order to prevent biasing our initial assessments of our method, we produced this figure blindly, with the identity of the correct match unknown to the operator of the shotgun mapping application. We found that one match score fell far below the mean of all the match scores (5 sigma away), and was significantly lower than even the next best match score. At this point, we unblinded the file number of the correct match, the pBR322 simulation and confirmed that our algorithm successfully identified the experimental fragment, based on the criteria of best match score.

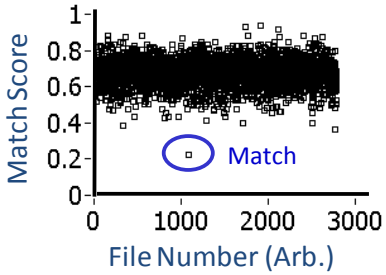


Figure 2.3. Compilation of match scores for a single experimental data set. The file number is an arbitrary, arising from the order in which the library simulations were loaded. A perfect match would have a score of zero, and the correct match can be seen as having the lowest score, very distinguishable from the incorrect matches.

Robustness Analysis

Fig. 2.3 shows successful shotgun DNA mapping for one of the experimental data sets. We repeated this for all 32 data sets and the correct match was the best score in every case. We did not find any instance of incorrect assignment for the window size and location we chose. (Some window sizes and locations produced failures.) To better visualize the robustness, we created histograms of all the scores for all the matches (N=32) and all the mismatches (N≈2700*32) and fit these histograms to Gaussian functions. These data are shown in Fig. 2.4., with the correct matches in blue and the mismatches in red. The integrated area of overlap between the two Gaussian fits is a small number, another indicator of the expected rate of false positives. The only overlap is in the tails of the Gaussians, a region where it is likely that the true experiments would significantly differ from a normal distribution, so this only provides an estimate of the true error rate.

The robustness shown in Fig 2.4 is somewhat surprising, given the effect of viscous drag on the experimental unzipping forces. We found that the match scores relative to the mismatches was not much different for these data sets, compared to one data set we obtained without the viscous drag effect (data not shown). A possible explanation for this is that the pBR322 sequence has high

GC content in the comparison region, and thus a vertical shift of the data merely tends to shift both the correct matches AND the mismatches to higher values, without increasing the overlap of the two histograms shown in Fig. 2.4.

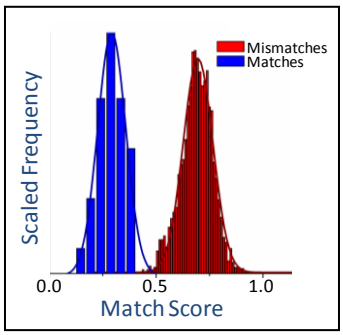


Figure 2.4. Comparison of 32 match scores to all mismatch scores. Blue histogram represents the match scores for the 32 experimental data sets, while red histogram (will) represent all incorrect match scores. *Solid lines* are fits to the normal distribution. Overlap of the two distributions indicates probability of false positives.

These results give us confidence that we will be able to perform SDM of yeast genomic DNA. It is in this application that we will use XhoI fragments, unlike in this proof-of-principle where we have used existing pBR322 unzipping data. The reasoning behind the use of published pBR322 is that the facilities for unzipping XhoI-digested yeast DNA is unavailable to us at this time. It is possible that the pBR322 sequence has special features that may cause us to overestimate the likelihood of success of the SDM method. We fell this is not likely the case for two reasons: (1) we do not see anything special in the pBR322 sequence by eye and (2) we obtained promising results from other experimental data, namely from pCP681 which is an unrelated, highly repetitive plasmid (data not shown).

Future Improvements

Based on our results, we expect SDM will work well for 6bp recognition sites in a genome the size of yeast. It is not clear how well it will work for shorter recognition sites or larger genomes, both of which will produce much larger libraries

(for example XhoI sites in the human genome). There are many independent avenues of optimization which gives us great confidence that this will work for these much larger libraries. These opportunities include: data acquisition, data processing, improved simulation algorithms and matching algorithms.

One very promising avenue is to improve the simulation by including the base stacking interactions (SantaLucia 1998; Bockelmann *et al*. 2002) and elastic energy of the dsDNA anchor. These known systematic errors in the existing simulation do not currently inhibit the function of the simulation. However, elimination of these errors will allows us to work with much larger libraries and matching windows much closer the initial unzipping point.

Improving the matching algorithm is another promising and independent avenue for optimization. Currently we have a simple algorithm based on the standard deviation between the two curves. There are clear opportunities to explore more advanced manipulations of the data in order to improve the signal to noise ratio. First, optical tweezers data can have slight length errors due to microsphere size variation, drift, or other causes. We can account for this possible shift in our algorithm by allowing for stretching of the data sets. Next, we can develop an independent match criteria that when combined with the current criteria dramatically increase the signal to noise ratio. This may include Fourier space manipulations such as cross-correlation which we found ineffective on its own but may add value in combination with the current match criteria.

Conclusions and Future Work

SDM looks like a very promising avenue for a new mapping technique. We believe it may work with 6 base pair recognition sequences in yeast DNA. We also believe this technique can aid in telomeric studies since the repetitious nature of telomeric sequence would be easy to identify using unzipping forces. We also may be able to quickly identify alternative spliced DNA sequences that can in turn be used to cancer research. Our initial idea of using SDM to study DNA replication devices in chromatin is a very exciting possibility. However, we first need to show that SDM works experimentally. In this regard we are attempting to digest yeast genomic DNA using XhoI. We then will attempt to identify cloned sequences from the genome as proof that SDM works.

Acknowledgments

The pBR322 experimental unzipping data used in this paper were obtained by SJK in the laboratory of Michelle D. Wang and were used for equilibrium constant measurements published in 2002 (Koch *et al*. 2002). The optical tweezers instrumentation and control and analysis software were constructed by authors on that publication and Richard C. Yeh. We thank the following people for useful discussions, some of whom may end up being authors on future versions of this paper: Mary Ann Osley, Diego Ramallo Pardo, Kelly Trujillo, Toyoko Tsukuda, Karen Adelman, David J. Clark, Mike Pikaart, Brandon Beck, and many anonymous grant reviewers. Diego Ramallo Pardo and Brandon Beck have performed work towards obtaining native chromatin and DNA constructs for shotgun DNA mapping of yeast genomic DNA. LJH and ALS were supported by a grant from the American Cancer Society and NSF IGERT Fellowships. RAM supported by a

grant to the CHTM from Emcore corp. SJK and LNL supported by start-up budget from the CHTM.

Discrete State Model for Kinesin-1 with Rate Constants Modulated by Neck Linker Tension

Lawrence Herskowitz and Steven Koch

Introduction

Kinesin is a family of motor proteins that catalyzes ATP hydrolysis and steps along a microtubule via a series of stochastic transitions (Cross et al. 2000; Coppin et al. 1997; Svoboda et al. 1993). Kinesin-1 (herein referred to as simply "kinesin") is an often-studied member of this family. It has an essential role in anterograde axonal transport (Muresan 2000; Nakata and Hirokawa 2003; Martin, Hurd, and Saxton 1999; Goldstein and Philp 1999; Goldstein and Yang 2000; Dennis, Howard, and Vogel 1999; Duncan and Goldstein 2006). Kinesin can walk about a micron along the microtubule in a second. It does this through a cycle that involves hydrolyzing one ATP per step. This stepping cycle has been probed extensively through many different experiments and tools including optical traps, analysis of chimera from different kinesin family members, Förster resonance energy transfer (FRET), and gliding motility assays (Ma and Taylor 1995; Adio *et al*. 2009; Block *et al*. 2003; Wagenbach, Coy, and Howard 1999; Gilbert *et al*. 1995; Rosenfeld *et al*. 2002; Verbrugge, Lansky, and Peterman 2009). This has led to an understanding of the frequent transitions that under normal conditions the kinesin steps through in order to travel along the microtubule. Details of the ordering of the transitions are still debated. For example, the exact order of ATP binding and the mechanical step of kinesin moving forward are argued, and these steps are sometimes combined (Guydosh and Block 2006).

There are a couple of important characteristics of the kinesin stepping cycle. The first is that one ATP is hydrolyzed per step, providing energy for directed transport along the microtubule (Wagenbach, Coy, and Howard 1999; Schief and Howard 2001). Under normal conditions ATP binds to the head and is hydrolyzed into ADP and inorganic phosphate (P_i) which are then released. Another important feature is that there is coordination between the two heads. The neck linker domains can transmit strain which is modulated by the nucleotide binding states of the two heads. This coordination allows for high processivity. Kinesin can walk hundreds of steps before detaching from the microtubule.

The neck linker has been reported to be within a range of 14-15 amino acids (aa) long or 5.32-5.7 nm, assuming a contour length of .38 nm per aa (Shastry and Hancock 2010; Miyazono *et al*. 2010; Muthukrishnan *et al*. 2009; Yildiz *et al*. 2008). The neck linker follows the C-terminal of the catalytic core (Hahlen *et al*. 2006; Rice, Lin, Safer, CL Hart, Naber, Carragher, Cain, Pechatnikova, Wilson-Kubalek, Whittaker, Pate, Cooke, Taylor, Milligan, and Vale 1999a). It is believed that coordination of the heads is assisted by a nucleotide dependent docking mechanism of the neck linker. When ATP or ADP-P_i is bound to the catalytic core, the neck linker will become more ordered and bind to the head while extending towards the plus end of the microtubule. This docking to the head occurs when a short amino acid sequence on the head denoted switch I chemically interacts with the P_i (Rice, Lin, Safer, Hart, Naber, Carragher, Cain, Pechatnikova, Wilson-Kubalek, Whittaker, Pate, Cooke, Taylor, Milligan, and Vale 1999b; Shastry and Hancock 2010; Sindelar *et al*. 2002; Vale and Milligan 2000a; Yildiz *et*

al. 2008; Muthukrishnan *et al*. 2009; Miyazono *et al*. 2010; Guydosh and Block 2009). Conversely the neck linker is found in a less ordered or undocked state when the head is empty or bound to ADP. Though this docking mechanism has been extensively explored it is not known exactly how or if the neck linker coordinates a long processive kinesin motion.

Many researchers have developed models to explain kinesin's processivity, force generation, and other physical aspects. These models include ratchet models (Fan et al. 2008; Z Wang et al. 2007; Astumian and Derényi 1999), elastically coupled Brownian heads (Derényi and Vicsek 1996; Kanada and Sasaki 2003; Peskin and Oster 1995; Shao and Gao 2006; Thomas et al. 2002), and discrete-state stochastic models(Shastry and Hancock 2010; Muthukrishnan *et al*. 2009; Gilbert, Moyer, and Johnson 1998a; Fisher and Kolomeisky 2001; Kolomeisky and Fisher 1999; Liao *et al*. 2007; Gilbert, Moyer, and Johnson 1998b). While these models have produced invaluable insight into kinesin, there are still many unanswered questions especially concerning how the neck linker physically impacts kinesin's behavior and how the behavior changes under various conditions.

In this paper we describe a discrete-state model of kinesin, analyzed by stochastic simulation and analytical Markov chain theory. There are two aspects of our model that have not been typically included in prior work. First, we do not impose strong restrictions on allowed transitions between states, even if these states are considered forbidden or rare. This allows us to analyze rare transitions, in particular those leading to two-headed detachment. On the other hand,

our permissive model requires knowledge of many more rate constants beyond those on a core cycle. The second feature we include is a physics-based model of how the kinesin neck linker strain modulates rate constants. We model the neck linker as a worm-like chain (WLC) and use Kramers' reaction rate theory to model force-based modulation of head binding and unbinding rates. We also include explicit chemical gating based on potential neck linker strain.

In this paper we describe the numerous rate constants we used in our model and our methods for estimating or calculating unknown rate constants. The rate constants may serve as a useful review of existing published rate constants for kinesin-1. We demonstrate that our model produces results agreeing with many published experiments using both Monte Carlo and Markov chain analysis. Finally, we explore how observables such as speed and processivity are affected by changing the neck linker length and thus the strain. The software applications presented have been written in LabVIEW 7.1 and are available as open-source on SourceForge at http://sourceforge.net/projects/herskowkinesin/files/.

Methods

Model Description

As mentioned above, we have created a permissive model that allows for rare states and transitions and does not explicitly define a core cycle. The model is defined by a vector of possible two-headed states and a matrix of transition rates. An individual head can be either bound or unbound to the microtubule and can have four nucleotide states (ATP, ADP-P_i, ADP, or no nucleotide). We have limited the nucleotide state to not allow binding of P_i by itself. We do this to reduce

complexity of the model and are motivated by the low concentration of P_i in solution and the lack of discussion of this state in the literature (Arnal and Wade 1998; Kawaguchi and Ishiwata 2001; Vale and Milligan 2000b). Two-headed states can be defined as combinations of one-headed states. When both heads are bound, however, an additional property defines relative front/back position of the heads. When one or both heads are unbound, there is no front / back property.

This results in 80 unique two-headed states. The transition rate matrix has a size of 80 x 80, but with only 6 or seven potentially non-zero entries per row. This is because transitions are restricted to one chemical reaction or binding event at a time. For example, an ATP head can transition to an ADP-P_i head (hydrolysis) or an empty head (ATP release), but cannot transition to an ADP-only head, since that would require two simultaneous events: either ATP release followed by ADP binding, or hydrolysis followed and P_i release. This results in a sparse matrix with 416 allowed transitions. Of these, there are 148 unique transitions. These rate constants are the core of the model. In the following sections, we discuss our methods for obtaining these rate constants, which includes the literature, mechanical and chemical gating, and empirical fitting. In supplemental table S1 located at http://openwetware.org/wiki/User:TheLarry/Notebook/Dissertation_Files#Rate_Constant_File we provide a list of all the transition rates we used for the data shown in this paper. When available, we provide the published range and when necessary our reasons for the value we have used.

Use of Published Rate Constants

The first thing we did was to perform an extensive search of the literature for well-accepted published rate constants. Since kinesin is well studied, we were able to find rates for many of the transitions (Bustamante *et al*. 2004; Carter and Cross 2005; Carter and Cross 2006; Fan *et al*. 2008; Farrell *et al*. 2002; Gilbert, Moyer, and Johnson 1998a; Gilbert *et al*. 1995; Hackney 1996; 1988; Hancock and Howard 1999; Hyeon, Klumpp, and Onuchic 2009; Imafuku, Thomas, and Tawada 2009; Kaseda, Higuchi, and Hirose 2003; Liu, Todd, and Sadus 2005; Ma and Taylor 1997; Rosenfeld *et al*. 2003; Rosenfeld *et al*. 2002; Schnitzer, Block, and Visscher 2000; Shao and Gao 2006; 2007; Thorn, Ubersax, and Vale 2000). Of the 148 unique rate constants 64 were taken from literature. This number does not account for the 8 inorganic phosphate release rates explained in the Empirical Fitting section below.

Hancock *et al.* hypothesized that rate constants for a singly-bound head are similar to those for monomeric kiensin constructs (Hancock and Howard 1999). We used this reasoning for 12 unique transition rates extracted from the literature (Fan et al. 2008; Gilbert, Moyer, and Johnson 1998b; Ma and Taylor 1997b; Cross 2004). This included the nucleotide-dependent head unbinding rates that are used to set the values of k_0 in equation 3.2.

Besides the inorganic phosphate release rates described below, we set the rate constants within the published ranges we found. We did not pick the center of these ranges, but instead made some adjustments to produce results for run length and run time that agreed with predictions. Unfortunately we did not record our exact method for doing this during the initial phases of our research, so we

cannot describe it completely here. In many cases, we agreed with a specific reference, and in Table S1 we so designate these rate constants. However, we cannot state our specific reasons for choosing the particular reference.

Neck Linker Modulation of Rate Constants

Using an extensible WLC model, we can approximate the tension in the neck linker in different docking configurations. Calculating this tension allows us to modulate the rate for a bound head to detach from the microtubule using the Bell equation (Bell 1978). It also allows us to calculate the rate at which an unbound head can bind to the next and previous binding sites. These two rate modulation methods are explained in the Mechanical Strain Gating section. Finally, we hypothesize that when the tension in the neck linker is great, switch I would not be able to bind to the inorganic phosphate, thus the neck linker would not be able to dock. This prevents a bound head from entering a docked state which increases the nucleotide release rate. This is explained in the Chemical Strain Gating section.

Mechanical Strain Gating

We model the neck linker as an extensible WLC. The following interpolation equation describes the relationship between the force and the extension of a WLC (M Wang et al. 1997)

$$\frac{FP}{k_B T} = \frac{1}{4}\left(1 - \frac{x}{L_0} + \frac{F}{K_0}\right)^{-2} - \frac{1}{4} + \frac{x}{L_0} - \frac{F}{K_0} \tag{3.1}$$

where F is the force in pN, x is the extension length in nm, P=0.8 nm is the persistence length (Miyazono et al. 2010; Z Wang et al. 2007; Hariharan and Hancock 2009), k_BT=4.1 pN-nm is the thermal energy, K_0=1000 pN is the stretch

modulus (Cui and Bustamante 2000; Van Leeuwen 2006), and L_0 = N*0.38 nm is the contour length, dependent on the number of undocked amino acids (N). We use N=26 as the number of amino acids in the two fully-undocked neck linkers of the dimer, for L_0 = 9.88 nm. The 13 amino acid length of a single neck linker is within the published range, 12-17 aa (Shastry and Hancock 2010; Miyazono et al. 2010; Muthukrishnan et al. 2009; Tomishige, Stuurman, and Vale 2006; Kikkawa 2008; Asbury 2005), of neck linker lengths. We use this model, as opposed to a hookean approximation because the hypothetical forces encountered from the various docking states are well outside of the low-force linear regime.

When one of the neck linkers goes from an undocked to a docked configuration, two parameters change. First, the number of free amino acids between the two kinesin heads changes by the amount of docked amino acids. This number is debated; for this work, we use 10 amino acids (Sindelar et al. 2002; Vale et al. 2000; Case et al. 1997)The second parameter that changes is the position of the attachment of the free amino acid chain to the kinesin head. We use a highly-simplified one-dimensional model for the docking and assume that the attachment position shifts towards the plus end of the microtubule by the number of amino acids multiplied by the contour length per amino acid. This effectively assumes that there is enough binding energy to stretch the neck linker to its contour length, a force of about 109 pN for the WLC parameters we use. If both heads are bound, neck linker docking or undocking either reduces or increases the strain between the two bound heads, depending on whether the change in

docking state occurs in the front or rear head. See Figure 3.1 for the strain forces in the various docking states.

The mechanical strain affects our model in two ways. The first is by modulating the rate constants for head unbinding from the tubulin binding site. Strain increases the unbinding rate exponentially according to the Bell equation (Bell 1978).

$$r = \frac{k}{k_0} = e^{\frac{\int_{l-\delta}^{l} F(x)dx}{k_B T}} \tag{3.2}$$

where k_0 is the unbinding rate with no force applied, l is the distance from the unbound equilibrium position to the position at its binding site, and δ is the distance from the binding site to the transition state. δ has been measured experimentally to be 2.5-3 nm (Block et al. 2003; Block 2007; Kawaguchi, Uemura, and Ishiwata 2003) and we use 2.5 nm in this work. We assume that δ is the same whether force is applied by the neck linker or an external force probe, and that it is constant over the force range. Figure 3.1 shows the unbinding rate increase factors for the various docking states. We were able to calculate 32 unique detaching rates using equation 3.2.

The second way neck linker strain affects rate constants is by modulation of the rate of head binding to tubulin binding sites. We first made the assumption that when the head detaches from the microtubule it does not rebind before quickly finding an equilibrium position over the bound head. This rate of reaching equilibrium can be estimated from the Smoluchowski equation and the WLC forces. It is of the order of 10^9 s^{-1} and rapid enough to ignore in our model. Neck

linker docking of the heads affects the equilibrium position of the undocked head relative to its forward and backward tubulin binding sites, as illustrated in Figure 3.2. The equilibrium position affects the rate constant for binding to the forward or backward binding site. We model this landscape as a WLC potential coupled with an absorbing potential, which creates a cusp. The minimum of the energy landscape is placed at head equilibrium location. Hanngi, et al. 1990 (Hänggi and Borkovec 1990) reports the rate for a particle to escape a cusp shaped barrier.

$$k_{cusp} = \frac{DU''(x_0)}{2\pi k_B T}\sqrt{\frac{\pi E_b}{k_B T}}e^{\frac{E_b}{k_B T}} \tag{3.3}$$

In the previous equation k_{cusp} is the escape rate to get over the barrier, D is the diffusion constant, $U''(x_0)$ is the second spatial derivative of the energy landscape evaluated at the energy minimum, and E_b is the cusp barrier height. In this case E_b is the energy of the WLC at the distance to the binding site. The distance of the cusp from the equilibrium position is equal to the distance to the forward or backward binding site from the equilibrium position. Thus, the magnitude of E_b depends strongly on the docking states of the bound and unbound heads. It is worth noting that this equation depends strongly on E_b and less on the curvature at the equilibrium position, but does not depend on the energy potential after the cusp. In addition to the assumptions already described, we also explicitly assume that there is no reduction in binding rates due to configuration of the binding interfaces, for example, rotation of the head relative to the binding site. That is, binding to the microtubule is instantaneous once the head diffuses a dis-

tance equal to the distance from the binding site. Using equation 3.3 we were able to calculate 32 unique stepping/binding rates.

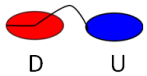
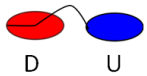
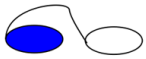
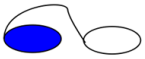
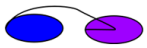
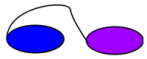
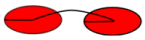
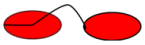
Unrestricted Configuration	Unrestricted Tension (pN)	Configuration used in model	Tension used in model (pN)	Contour length used in model (nm)	Extension used in model (nm)	r
 D U	17.1	 D U	17.1	6.08	4.4	109.4
 U U	33.6	 U U	33.6	9.88	8.2	$3.5x10^4$
 U D	1041	 U U	33.6	9.88	8.2	$3.5x10^4$
 D D	2631	 D U	17.1	6.08	4.4	109.4

Figure 3.1. Docking configurations for the two head bound states and their associated tension, neck linker contour length and extension, and head detachment rate enhancement factor. Color coding indicates nucleotide state of the head: white no nucleotide, red ATP, purple ADP-P_i, blue ADP. There are a total of four possible docking configurations for two head bound states. These are seen in the "Unrestricted Configuration" column. We modeled the neck linker using an extensible worm-like chain, Equation 3.1, with a persistence length of 0.8 nm, thermal energy value of 4.1 pN-nm, elastic modulus of 1000 pN/nm, and a varying neck linker contour length and extension. After calculating the tension in each unrestricted configuration we came to the conclusion that the neck linker could not stably dock in the front head when there is a high force pulling backwards as seen in the undocked/docked and docked/undocked configurations. Though not shown in the table the neck linker contour length and extension for undocked/docked is 6.08 nm and 12.0 nm respectively. For docked/docked the parameters are 2.3 nm and 8.2 nm respectively. Thus in our model prohibitive tension causes the front head to be in an undocked state regardless of nucleotide status. The changes this has on the configurations can be seen in the column labeled "Configuration used in model". The first two rows are allowed while the last two rows show prohibited front head docking. The resulting tensions from the actual configurations used are shown in the fourth column. Finally the unbinding rate enhancement factor, r, which is the ratio of the kinesin's head detachment rate with the tension compared to the rate with no force applied (calculated using Equation 3.2) is shown in the last column.

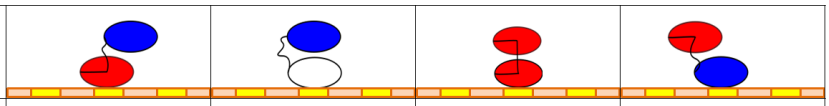
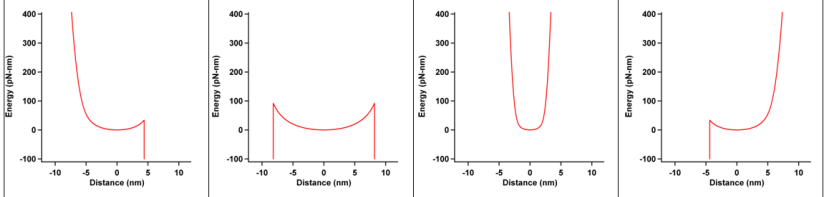
Configuration				
Contour Length (nm)	6.08	9.88	2.28	6.08
Energy Landscape Diagrams				
Distance to Binding Site (nm) Backward / Forward	12 / 4.4	8.2 / 8.2	8.2 / 8.2	4.4 / 12
Rate for Binding (s^{-1}) Backward / Forward	0 / 4.9×10^5	36.8 / 36.8	0 / 0	4.9×10^5 / 0

Figure 3.2. Forward and backward binding rates for an unbound head for differing two-headed docking configurations. When a head detaches from the microtubule it quickly finds an equilibrium position over the bound head. This is because the tension in the neck linker is a minimum in the least extended position. The four different two-headed docking configurations are shown in the table. Unlike Figure 3.1, all docking configurations are allowed, since only one head is bound and there is no neck linker tension. This equilibrium position is the minimum of a potential energy landscape determined by the worm-like chain behavior of the flexible (undocked) portions of the two neck linkers. To model binding, we placed a cusp at the location of the forward and backward binding sites as seen in the "Energy Landscape Diagrams" row. Even though each landscape should have two cusps, in most cases the energy becomes exceptionally large and effectively prohibits binding in a direction, forward or backwards. The rates for reaching the forward and backward binding sites are computed using Equation 3.3 using a diffusion constant of 5.05×10^8 nm/s^2, 4.1 pN-nm thermal energy, and the same parameters for the worm-like chain mentioned above. E_b is the energy value from the worm-like chain potential at each binding site. When the contour length is too short or the distance to the binding site too large, the rate for reaching a binding site effectively becomes zero. Note by putting a cusp at the binding site, we are modeling binding as immediate if the molecule reaches that extension. This is the simplest model, and we do not account for the need for correct 3-D orientation of the binding sties or other factors. Color coding and neck linker representation same as in Figure 3.1.

Chemical Gating

In addition to head binding and unbinding rates, neck linker docking may also affect nucleotide binding stability. ADP binds more weakly to the head than ATP or ADP-P_i (Carter and Cross 2006; Cross 2004). This could be due to stabilizing interactions between the switch I and the inorganic phosphate (Kikkawa et al. 2001).

According to the parameters used above in Figure 3.1, the tension in the neck linker when an undocked head is bound behind a bound docked head would be

1041 pN, and when both bound heads are docked the tension is increased to 2631 pN. Such high tensions in the neck linker and consequently on the switch I, would likely prohibit the switch from binding to the inorganic phosphate thus the neck linker would not be able to dock to the head. Because it is highly unlikely that there is enough binding energy to sustain this force we make the assumption that these docking configurations are forbidden, and therefore the inorganic phosphate cannot bind to switch I. Because of this, binding of ATP is not stabilized and its unbinding rate from the kinesin head is the same as ADP's unbinding rate. In most cases, our gating of the nucleotide unbinding rate is in contrast to much chemical gating literature, where gating modulates the nucleotide binding rate (Guydosh and Block 2006; Muthukrishnan et al. 2009; Shastry and Hancock 2010). In some cases, we use specific published rate constants which may imply strong chemical gating of nucleotide binding rates. For example our ATP binding rates of 3 μMs^{-1} when the empty head is in front and 0.3 μMs^{-1} when the empty head is in back (Shao and Gao 2006).

Empirical Fitting of Inorganic Phosphate Release Rates

Without changing the rates of inorganic phosphate release well outside of published experimental ranges, we were unable to produce results that matched expected run time and run length. This same problem was encountered by the model of Muthurkrishnan et al. (Muthukrishnan et al. 2009) and Shastry et al. (Shastry and Hancock 2010). The literature reports a range of 13 to 100 s^{-1} (Gilbert, Moyer, and Johnson 1998a; Farrell et al. 2002; Shao and Gao 2007; Liu, Todd, and Sadus 2005). However, release of inorganic phosphate must occur

during every productive hydrolysis cycle, which occurs at a rate of 100 s^{-1}. Shasta et al. point out that this is incongruous with such a low rate of phosphate release. They empirically adjust their rate to 250 s^{-1} for the step of bound/ADP-P_i in back of a bound/empty head (Shastry and Hancock 2010). We arrived at a similar conclusion and needed to adjust our rate to 250 s^{-1} for all of the release rates except when ADP-P_i is behind an undocked head. For the cases of an ADP-P_i head bound behind a bound ADP or empty head, we used a P_i release rate of 25 s^{-1}. This is in range of published rate constants but we don't have a structural reason for using the published range for only these particular constants. However, we found that if the rate constant for phosphate release behind an ADP head were also 250 s^{-1} it would frequently enter the ADP/ADP state, which leads to a reduction of processivity. We use 25 s^{-1} for phosphate release behind the empty head as well, reasoning that the chemical gating should be the same for both cases. In the case of Shastry *et al.* (Shastry and Hancock 2010), they only consider phosphate release from behind a bound empty head. For consistency, we kept the inorganic phosphate release at 25 s^{-1} when in back of an empty head, even though it didn't affect our run time or length.

Agent-Based Stochastic Simulation

We used an agent-based implementation of the model for stochastic simulation (Gillespie et al. 2009; Gillespie 1976). Each kinesin head is an agent identical to the other head and handled independently. Each agent is a state machine with 8 states—bound or unbound in each of the 4 nucleotide states. We initially thought this would reduce the complexity of the simulation, using a state machine

with only 8 states instead of 80 if we considered both heads at once. However the need to modulate rate constants based on the state of the partner agent required many nested case structures for each state, and thus we did not realize a gain in simplicity. Identical results should be obtained with an 80 state machine, but we did not attempt to show this. As an example, Figure 3.3 shows the possible transitions from a bound/ATP head with an unbound/ADP head. The ATP head can unbind from the microtubule, hydrolyze to form ADP-P_i, or release ATP. The other head can release ADP, bind behind the ATP head, capture inorganic phosphate, or bind in front of the ATP head. The most likely transition is for the unbound head to bind in front of the ATP head. The ATP head has a binds relatively strongly to the microtubule so it is unlikely to unbind from the microtubule. While hydrolysis is fast and expected in other two headed states, in this case it is three orders of magnitude slower than the forward stepping rate. The same is true for ATP release from the bound head. ADP is strongly attached to an unbound head (Cross 2004). Inorganic phosphate is unlikely to bind to the unbound head with its extremely low concentration, and it is unlikely for the unbound head to bind to the previous binding site since it has to travel 12.0 nm as opposed to a 4.4 nm distance to the next site. This makes a huge difference in the rate of forward or backward binding as seen in Figure 3.1.

We used a Monte Carlo method to determine which path the kinesin will take from the current state. A random number, rand, between 0 and 1 is chosen and converted to an exponentially distributed time, t_i, according to the following equation, $t_i = -\frac{1}{k_i}\ln(rand)$, where k_i is the rate constant for the ith transition. The sys-

tem is moved to the state according to the transition with the shortest time (Gillespie 1976). The process is repeated for the new state, with new rate constants. If the transition is a head unbinding or binding event, then the stalk position moves forward or backward according to neck linker docking configurations.

For the work reported here, each single run starts with the system in unbound ADP with a bound ATP, and ends when both heads are unbound simultaneously. The states of the heads, stalk positions, and cumulative time are recorded for each transition. Most reported results are the result of an average of 1000 individual runs per condition. This application is called Kinetic Monte Carlo.exe available at http://sourceforge.net/projects/herskowkinesin/files/.

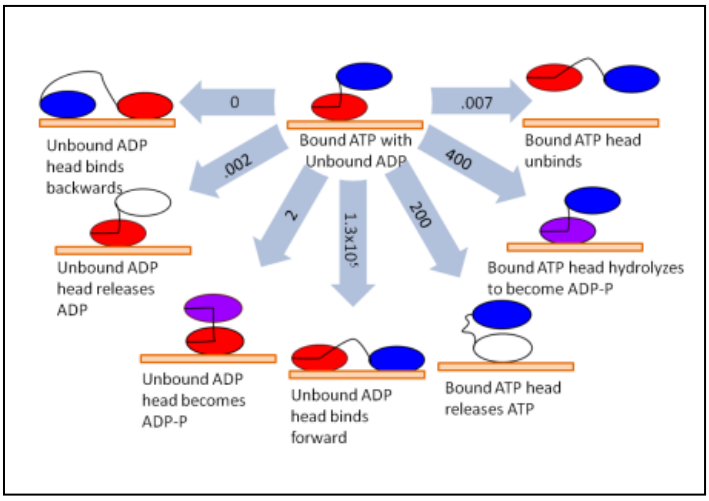


Figure 3.3. Permissive model transitions from an ATP head bound to the microtubule and an ADP head unbound (top, middle). Color coding indicates nucleotide state of the head: white no nucleotide, red ATP, purple ADP-P_i, blue ADP. Rates are in units of inverse second and are described in the text. The most likely transition is forward binding of the ADP head, due to the docked neck linker (black horizontal line) on the ATP head

Markov Analysis

Using a kinetic Monte Carlo technique is the simplest way to analyze this model, and it allows for measuring the variance of observables. Complementary and exact solutions for average run length, most frequented states, most probable transitions, run time and other observables can be obtained from Discrete

Time Markov chain (DTMC) and Continuous Time Markov chain (CTMC) theories.

DTMC

To calculate run length we analyze the DTMC that is embedded in the full CTMC. The embedded DTMC is calculated through

$$p_{ij} = \begin{cases} \dfrac{q_{ij}}{\sum_{j \neq i} q_{ij}} & j \neq i \\ 0 & j = i \end{cases} \tag{3.4}$$

where q_{ii} is the transition rate from state i to state j (Stewart 1994). The diagonal terms are 0 since the state cannot make a transition from state i to itself. We use the DTMC instead of the CTMC because we are only interested in the number of steps taken. It is simpler to use DTMC for this purpose since we are not interested in the time it takes which would require CTMC modeling.

To compute the average distance the kinesin travels, we first needed to calculate the probability that the system will return to the any state after n steps. This probability is expressed by

$$f_{ij}^{(n)} = p_{ij}^{(n)} - \sum_{l=1}^{n-1} f_{ij}^{(l)} p_{jj}^{(n-l)} \tag{3.5}$$

$f_{ij}^{(n)}$ is the probability that the first time the system enters state j after starting in state i is after n steps. $p_{ij}^{(n)}$ is the i,j element of the embedded DTMC raised to the nth power (Stewart 1994). So to calculate the probability that the system will ever transition into state j we use equation 3.6.

$$f_{ij} = \sum_{n=1}^{\infty} f_{ij}^{(n)} \tag{3.6}$$

Thus f_{ii^*} is the probability of starting in state i and ever transitioning to state i^*. i^* is the same state as i but with the opposite head in front. $f_{ii^*}^{(n)}$ is the probability to transition to the i^* n times (Stewart 1994). We made the assumption that each time the system travels from i to i^* the kinesin took a step. Thorn *et al.* reported a 99.3% chance of finishing a cycle after starting (Thorn, Ubersax, and Vale 2000). For the rate constants used in this report, DTMC analysis also showed a 99.3% chance of cycle completion.

DTMC analysis also allows calculation of the most-visited states and the most-popular transitions. First the probability matrix needs to be rewritten in canonical form. If there are *b* absorbing states (states where both heads are detached) and *m* transient states then the probability matrix takes the canonical form of

$$P = \begin{bmatrix} S & R \\ 0 & \mathrm{I} \end{bmatrix} \tag{3.7}$$

where S is an *m* by *m* matrix, R is an *m* by *b* matrix, 0 is an *b* by *m* zero matrix, and I is an *b* by *b* identity matrix. In our model there are 16 absorbing states (*b*=16) and 64 transient states (*m*=64). The fundamental matrix, N, is calculated by

$$N = (\mathrm{I} - S)^{-1} \tag{3.8}$$

where the superscript -1 denotes the inverse of $(\mathrm{I} - S)$. The elements of the fundamental matrix n_{ij} is the average number of times the system is in state *j* if it

started in state *i*. N, and P are used to calculate the most frequented states (Grinstead and Snell 1997).

To calculate the most popular transitions we take n_{ij} and multiply it by the probability, p_{jk}, to go from state *j* to state *k*. This allows us to find the average number of times each transitional step is taken.

Finally, we can use the R matrix to calculate the most probable absorbing state (completely detached state).

$$D = NR \tag{3.9}$$

where D is a matrix whose elements d_{ij} is the probability for a system whose initial state, *i*, will be absorbed in state *j* (Grinstead and Snell 1997).

CTMC

To calculate run time we used continuous time Markov chain (CTMC) analysis, since we needed to consider the time spent in each state. We first created the infinitesimal generating matrix, Q, out of the transition rates. The off-diagonal elements q_{ij} are the transition rates from state i to state j. The diagonals q_{ii} are calculated as

$$q_{ii} = -\sum_{i \neq j} q_{ij} \tag{3.10}$$

and represent the rate of staying in state i. Q was then used to create the differential equation:

$$\frac{dP(t)}{dt} = P(t)Q \tag{3.11}$$

where $P(t)$ is a matrix whose elements $p_{ij}(t)$ are the probability that starting in state i the system is in state j at time t (Stewart 1994).

The solution to this differential equation is

$$P(t) = P(0)e^{Qt} \tag{3.12}$$

where $P(0)$ is the initial state. This can be more simply evaluated using the eigenvalues and eigenvectors so that the solution is:

$$P(t) = P(0)Ae^{\lambda t}A^{-1} \tag{3.13}$$

where A is an eigenvector, A^{-1} is its inverse, and λ is its eigenvalue matrix. This has the advantage of it being simpler to calculate the exponential of a diagonalized matrix.

To calculate the run time, we looked at the probability of finding the system in a detached state as a function of time. There are 16 unique detached states; however both heads unbound with ADP is by far the most common, detaching by this route over 99.96% of the time. Thus it is a good approximation to consider only the probability of ending in the two-head unbound ADP state.

Using DTMC and CTMC analyses to probe other characteristics from this collection of transition rates proved to be increasingly complicated and difficult. The simplest and thus more convincing method is to use the stochastic simulation, though Markov chain analyses are quicker. However it is important to note that analytical Markov chain analyses may be useful for future work, including analysis of variance via maximum caliber methods (Stock, Ghosh, and Dill 2008; Ghosh and Dill 2006). All software used to analyze the data is called Markov Chain Analysis.exe available at http://sourceforge.net/projects/herskowkinesin/files/.

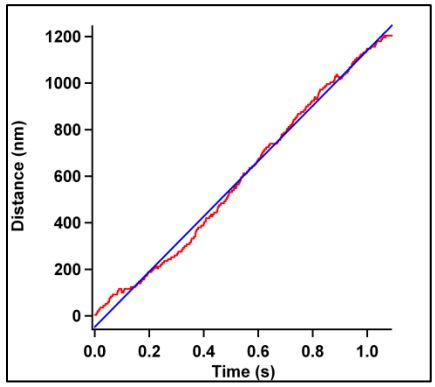


Figure 3.4. Example of distance versus time (red trace) for a single stochastic simulation. The linear least squares fit is shown as blue line and has a slope of 1187 nm/s.

To calculate velocity for a given run, we chose to use the best fit slope of the position versus time data, using least-squares fitting in LabVIEW 7.1. Seen in Figure 3.4 is an example of this best fit line over a single processive run. A group of 1000 runs will produce a spread of measured speeds as shown in Figure 3.5. We used a kernel density estimation (KDE) method to approximate the underlying probability density function (PDF) for that group of speeds (Silverman 1986). The peak of this PDF was used as the resultant speed for that set of conditions. For example each data point in Figure 3.8 is produced from the peak of the PDF for one hundred individual runs at a given set of conditions. KDE is an alternative to histogram methods for estimating PDFs. KDE is performed by summing up a kernel function centered at each data point, or mathematically written as

$$f_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{3.14}$$

where h is a smoothing parameter called the bandwidth, x_i are the data points, and $K\left(\frac{x - x_i}{h}\right)$ is the kernel function. We used the standard Gaussian kernel:

$$K\left(\frac{x-x_i}{h}\right)=\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-x_i}{h}\right)^2} \tag{3.15}$$

Figure 3.5 shows an example of the PDF produced from the KDE of 1000 speed measurements. For this work we used a high bandwidth of 200 nm/s because we are only concerned with finding a single peak as opposed to looking for speed changes or pauses. Smaller bandwidths sometimes produced multiple peaks which are not desirable for our purposes here. The higher bandwidth produced a larger spread than was intrinsic to the data, as seen in Figure 3.5.

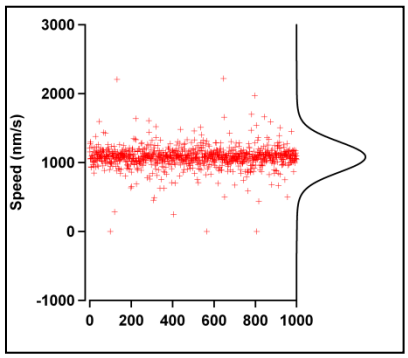


Figure 3.5. Speed for 1000 individual stochastic simulations (red crosses) and kernel density estimation (KDE) for speed probability density function (PDF, black curve). Each individual speed is the least squares fit slope of an individual run (see Figure 3.4). For KDE a bandwidth of 200 nm/s was used, in order to ensure a single peak in the PDF (see methods).

To calculate the Michaelis-Menten constant we best fit the speed versus concentration data using Igor Pro 5.05A (WaveMetrics Inc., Portland, OR) to

$$v([ATP])=\frac{v_{\max}[ATP]}{[ATP]+K_M}, \tag{3.16}$$

and we extract K_m, the Michaelis-Menten constant. $v_{\max}$ is the maximum velocity the kinesin reaches at saturating ATP concentrations. Igor Pro 5.05A uses a Levenberg-Marquardt algorithm to best fit the curve.

Extracting Observables from Markov Analysis

To calculate the expected run time and run length, we used the relation:

$$< x >= \int_0^\infty xP(x)dx \tag{3.17}$$

where x is either the run time or the run length parameter and P(x) is the probability density function from Markov theory. We evaluated this numerically with a dx of 8.2nm or .001 s.

Software for this Analysis

We used three different custom programs to produce the data seen in this paper. The workings of these programs have been discussed above. The names and summaries of their duties are listed below:

- Kinetic Monte Carlo
  - Allows the user to perform kinetic Monte Carlo simulations with prescribed rate constants.
- Analyze Simulations from Kinetic Monte Carlo Simulations
  - Analyzes a grouping of kinetic Monte Carlo simulations and extracts (all the following quantities are averaged over the group) run time, run length, ATP/step ratio, speed, most probable states, most probable transitions, most probable detached states, and view individual runs.
- Markov Chain Analysis
  - Uses Markov Chain theory to analyze prescribed rate constants. This software calculates run time, run length, most probable states, most probable transitions, most probable dethatched states, and creates the transition matrix.

These are provided as open sources at http://sourceforge.net/projects/herskowkinesin/files/. A tutorial video created by CamStudio is also available at this site called Kinesin Model Tutuorial.

Results

Reproduction of Widely-Accepted Experimental Results

Except when stated otherwise, the following parameters were used for all results reported below: 1000 micromolar ATP concentration, 100 micromolar ADP concentration, 0.1 micromolar P_i concentration, 8.2 nm tubulin dimer spacing, 4.1 pN-nm k_BT value. Neck linker properties are described in the methods. All analyses were started with the same initial state of bound ATP with an unbound ADP head.

As described in the method section, we adjusted some rate constants and neck linker properties to match correct run time and run length values. Figure 3.6 shows the histogram of the run length created from the kinetic Monte Carlo simulation in red with the function predicted by DTMC in black. The average run length of 1,298 nm is close to the range of published values of 600 to 1,200 nm (Vale et al. 1996; Thorn, Ubersax, and Vale 2000).

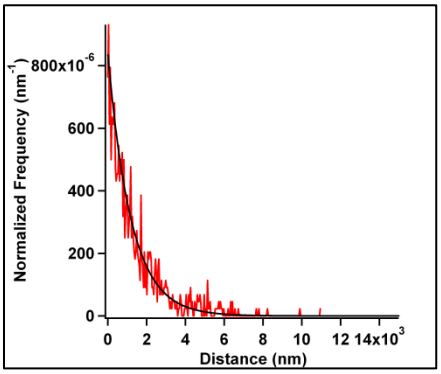


Figure 3.6. Processive run length results. A histogram of run length for the 1000 stochastic simulations is shown in red, while the calculated run length probability from DTMC is shown in black. The expected run length from DTMC is 1,298 nm and close to the published range of experimental averages.

The run time seen in Figure 3.7 has an average value of 1.31 s which is within the published range 0.75 to 2.89 s (Thorn, Ubersax, and Vale 2000; Verbrugge, Lansky, and Peterman 2009). The CTMC curve in black shows a sharp increase at 0 time. This is because kinesin cannot detach instantaneously from the initial state, but needs to cycle through other states before it can detach. We calculated a velocity of 1080 nm/s which is close to the range of the accepted experimental speed of 600- ~1,000 nm/s (Hunt, Gittes, and Howard 1994; Guydosh and Block 2009; Shastry and Hancock 2010).

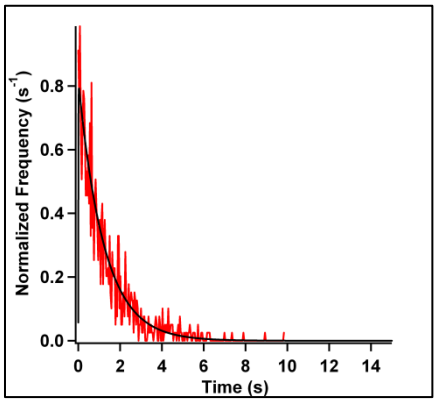


Figure 3.7. Processive run time results. Histogram of run time for 1000 stochastic simulations (red) is shown, along with calculated frequency from CTMC (black). The CTMC curve shows a minimum at zero time because more than one transition must occur to reach the most likely detached state (ADP bound to both heads), given our starting state (ATP head bound to MT, ADP head detached).

After setting rate constants and neck linker properties, we found that the ATP coupling ratio and ATP Michaelis-Menten constant were also in acceptable range. From the 1000 runs used in Figures 3.6 and 3.7, we found a ratio of ATP consumed to step taken of 1.03, and a ratio close to 1 is generally accepted (Schnitzer and Block 1997; Hua et al. 1997). To calculate the Michaelis Menten constant, we performed 100 simulations at each ATP concentration from 0 micromolars to 1,000 micromolar in variable increments. We computed the speed for each concentration, producing the Michaelis-Menten curve shown in Figure

3.8. The best fit Michaelis-Menten constant, K_m, was 37.6 μM, within the published range of 13-60 μM (Schnitzer and Block 1997; Hua et al. 1997; Verbrugge, Lechner, Woehlke, et al. 2009; Verbrugge, Lansky, and Peterman 2009).

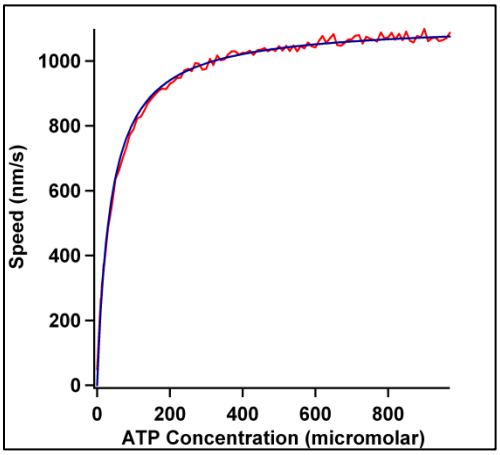


Figure 3.8. Speed versus ATP concentration from stochastic simulations. Each speed point (red curve) is the most likely speed found from kernel density estimation of 100 individual stochastic simulations. The blue curve shows the best fit to the Michaelis-Menten relation, with a K_M of 37.6 μM and a v_{max} of 1117 nm/s. This K_M is in the range of published experimental measurements.

Most Probable State for Complete Detachment

Using DTMC, we found a 99.96% chance of two-head detachment occurring with ADP bound to both heads. This result is not surprising since an ADP bound head has the weakest attachment to the microtubule (Asenjo, Weinberg, and Sosa 2006).

Core Cycle

One of our goals when developing this model was to see which core cycle(s) emerged constrained as much as possible by neck linker physics and published ranges without removing any possible states. Figure 3.9 shows the core cycles we found using the rate constants described here. We only include transitions that occur at least 10 times per processive run as computed from DTMC. None of the three ATP-turnover cycles seen in Figure 3.9 are the most common cycle de-

scribed in the literature, which involves an ADP head bound behind an empty, followed by ADP head unbinding and ATP binding to the empty bound head. A likely reason for this is that we do not inhibit ATP binding to the front head, regardless of docking state of rear head (Guydosh and Block 2006; Muthukrishnan et al. 2009; Shastry and Hancock 2010; Guydosh and Block 2009). This allows for states seen in the center and right of the figure, ATP-ATP and $ADPP_i$-ATP. In our model we did not forbid these states, we only forbid neck linker docking of front head in these states as described in methods. The outer cycle in Figure 3.9 is similar to Shastry 2010 (Shastry and Hancock 2010), which includes allows ATP binding to the $ADPP_i$ - empty state. However, other differences remain.

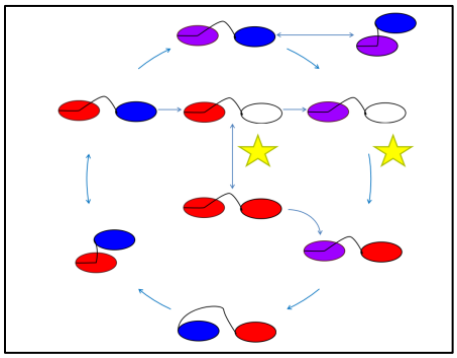


Figure 3.9. Most probable transitions resulting from our unconstrained model. For the main rate constants we used, we found the most likely transitions show here. Arrows indicate direction of transition. We only show transitions that occur at least an average of 10 times per processive run, as calculated by DTMC. Three core cycles can be seen in the picture, all proceeding clockwise. None of these are the cycle most commonly described in the literature, though we can recover that cycle by constraining the ATP binding to front head (marked by stars) as seen in Figure 3.10 and described in text. The nucleotide binding state of the heads are represented by color: white for no nucleotide, red for ATP, purple for ADP-P_i, blue for ADP. The neck linker is represented by the black lines, and docking is indicated by a straight horizontal segment, while undocked is curved upwards.

If we include strict front-head gating of ATP binding in our model, we can recover the more popular core cycles. When we reduced front head ATP binding drastically we found the core cycle shown in Figure 3.10. The rate constants we used for this core cycle can be found along with the open source software at http://sourceforge.net/projects/herskowkinesin/files/ called front “head gating rate

constants 2.dat". In order to get run length (514 nm) and run time (.92 s) in the correct range we needed to change the inorganic phosphate release in back of a bound empty head to 250 s^{-1}.

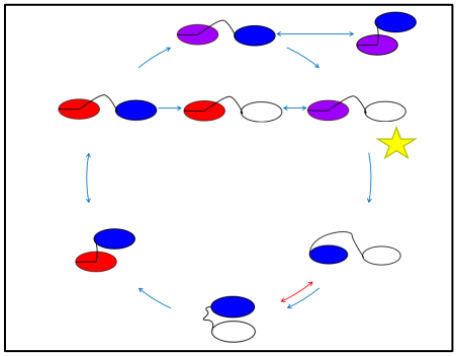


Figure 3.10. Most likely transitions after addition of front head gating of ATP binding. Only transitions that occur an average of 10 or more times per processive run (as calculated by DTMC) are shown. In contrast to Figure 3.9, here the rate constants for binding of ATP to the front head was forbidden if the rear head is in the ATP state. This is sometimes referred to as front head gating of ATP binding. In addition to reducing these rate constants close to zero, we also increased a rate of phosphate release to produce a reasonable run length and time (marked with STAR on figure, see text). Color coding and neck linker representation same as in Figure 3.1.

The core cycle for the low ATP and ADP case follows the same cycle as the front head gating except the unbound ADP head more frequently rebinds behind the empty head in the low concentration case. This is shown with the red arrow. Only transitions that occur an average of 10 or more times per processive run (as calculated by DTMC) are shown.

We also recovered popular core cycles by lowering the ATP and ADP concentrations but otherwise keeping the conditions similar to reported above. We lowered the ADP concentration because the processivity was too short with the 100 micromolar concentration we used for the other calculations and it is difficult to report a core cycle when the molecule does not take processive steps. With an ATP and ADP concentration of 3 micromolar, we found the core cycle seen in Figure 3.10. It is very similar to the case of front-head gating, except the unbound ADP head with the bound empty head more frequently rebinds behind the empty head (red arrow).

However, it was not our intention to reproduce the core cycle by modulating rate constants, but instead to see which cycles emerged from our rate constant

literature search and neck-linker modeling. Our open source software platform should allow other researchers to reintroduce this or other types of gating and explore the repercussions.

Changing the Neck Linker Length

A feature of our software is that it allows for easily investigating the effect of neck linker length on observables such as speed and processivity. We investigated these effects as we changed the neck linker length from 24 to 34 amino acids (26 amino acids was the standard length used). Figure 3.11A shows the processivity and Figure 3.11B shows the resulting speed. We observed a maximum in speed for the default neck linker length of 26 amino acids. The speed decreased by a factor of 2 when the neck linker was lengthened by 8 amino acids. The processivity steadily increased as the neck linker was lengthened, almost doubling with a neck linker change from 32 to 34 amino acids.

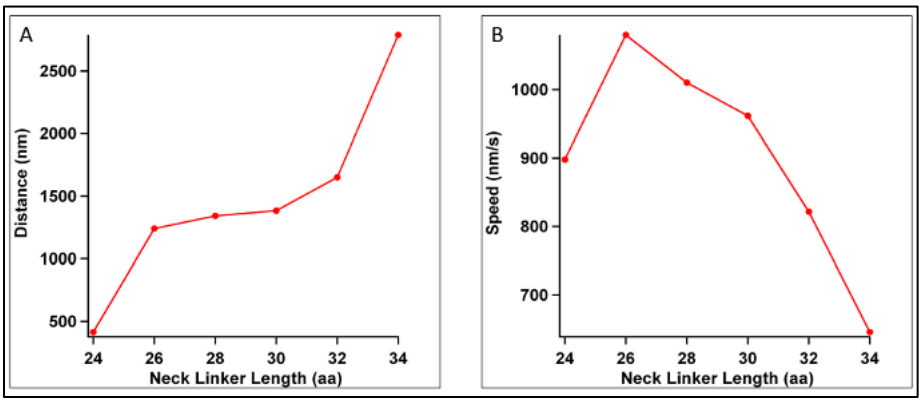


Figure 3.11. Effect of changing neck linker length on processivity (A) and speed (B). Changing the neck linker length changes the tension in the undocked portions of the neck linker. We did not adjust the number of amino acids involved in docking. The default neck linker length used throughout this report was 26 amino acids, with a docking number 10 amino acids.

In our model, increasing the neck linker length decreases the tension between the two bound heads. This decreases the rate at which a bound head de-

taches from the microtubule. On the other hand, it increases the rate the unbound head can reach a binding site. This decreases the speed by increasing the time of the two-headed states, but increases the processivity by increasing the chance of binding when unattached. The increased likelihood of backward steps also contributes to a decrease in speed and tempers the increase in forward processivity. The sharp decrease in processivity seen in the 24 aa neck linker can be attributed to the shorter neck linker increasing the unbinding rates drastically while decreasing the stepping rate. This causes the kinesin to spend most of its time with only a single head bound, causing a decrease in processivity. Table 3.1 shows the unbinding factor and stepping rates associated with each neck linker length.

Table 3.1. Effect of changing neck linker length on kinesin head unbinding and binding rates.

Neck Linker Length (aa)	Contour length (nm)	Docked contour length (nm)	r		Unbound Binding Rates (s^{-1})			
			Docked Undocked	Undocked Undocked	Docked Undocked		Docked Docked	Undocked Undocked
			4.4 nm	8.2 nm	4.4 nm	12 nm	8.2 nm	8.2 nm
24	9.12	5.3	$1.23x10^3$	$4.56x10^6$	$5.43x10^4$	0	0	0.184
26	9.88	6.1	109.445	$3.51x10^4$	$4.90x10^5$	0	0	36.7518
28	10.6	6.8	35.5	$2.58x10^3$	$1.28x10^6$	0	0	679.021
30	11.4	7.6	18.4	544	$2.15x10^6$	0	0	$4.05x10^3$
32	12.2	8.4	11.9	196	$2.93x10^6$	0	0	$1.34x10^4$
34	12.9	9.1	8.71	95.5	$3.59x10^6$	0	0	$3.13x10^4$

We changed the neck linker length from 24 to 34 amino acids (26 amino acids was used for all figures created in this paper unless stated otherwise). This table shows the contour length and docked contour length used to calculate r (the ratio of kinesin head detachment rate with a force compared to the detachment rate with no force) and the unbound head binding rates. As shown in Figure 3.1, there are only two different configurations when both heads are bound (although this assumption may become less valid as the neck linker length increases), and the neck linker extension is shown below each of these cases. For the unbound head binding rates, there are four different configurations. However, an unbound undocked head with a bound docked head has the same rates when traveling forward and backward as an unbound docked head with a bound undocked binding backward and forward. For simplicity we combined these into one column. The distances shown below each configuration is the distance to the closest binding site. Since docked/docked and undocked/undocked equilibrium positions are located exactly between two binding sites, the head needs to travel 8.2 nm to reach either binding site. r is calculated using Equation 3.2 while the unbound binding rates are calculated using Equation 3.3. As the neck linker length increases, the tension between the two bound heads decreases, thus r decreases as well. This means that the kinesin heads remain bound to the microtubule longer. The binding rates increase as the neck linker gets longer since there is now less force prohibiting the neck linker from diffusing that distance. However there still remains a bias for forward binding due to the forward bias in the equilibrium position. This causes a longer processivity since it decreases the time in the one-head bound state, thus reducing the probability of complete detachment. The velocity decreases due to the longer time spent with both heads bound to the microtubule.

Since our physics-based model of the neck linker is highly simplified, the quantitative results are less important than the qualitative trends. The trends of increased processivity and decreased speed with increased neck linker length do not agree with Yildiz (Yildiz et al. 2008) or Miyazono (Miyazono et al. 2010) or Shastry (Shastry and Hancock 2010). Disagreement with these experiments could be explained by a number of differences between our model and experimental conditions. In our model, as the neck linker length is changed, the mechanical gating changes dramatically—affecting head binding and unbinding rates. However, we do not include any effect of changing neck linker tension on chemical gating. The fact that we do not agree with experiment may indicate that chemical gating is significantly affected by neck linker length. It is also possible that our assumption of one dimensionality could cause these differences especially as the neck linker gets longer.

Summary

We developed an 80-state model for kinesin behavior that can be analyzed by stochastic simulation and Markov analysis. Unlike many existing models, we did not restrict the model to known core cycles. To do so, a large number of rate constants needed to be determined. We were able to set these rate constants by literature search and modeling of the neck linker for physical and chemical gating. To match experimental behavior, only the rate constant for inorganic phosphate release needed to be adjusted well outside of the published range. We were able to reproduce the expected results for run time, run length, speed, and processivity.

The advantage of our expanded-state model is that it allows for exploration of the behavior as rate constants are adjusted over a wide range, without the need to predefine a core cycle that may be changing over this range of parameters. We demonstrated an example of this as we explored the behavior as the neck linker length was changed. We also saw the core cycle change between high and low ATP concentration. The expanded-state model also allowed us to explore the most likely means of two-headed detachment. We expect to leverage this feature in future studies investigating the potential effects of osmotic stress and water isotope on kinesin processivity and speed. Finally, it is easy to limit the model and analyze published core cycles by setting the branch rates to zero.

We have begun work to add the ability to apply an external force to the kinesin. However this work remains complicated by the need to adjust many rate constants such as head unbinding, head rebinding, and chemical gating. Furthermore, forces add vectorially and depend on the location of force application.

All of the software used in this report is open source and available via http://sourceforge.net/projects/herskowkinesin/files/. A Tutorial video is available as well.

Acknowledgements

We thank Susan Atlas, Andy Maloney, Igor Kuznetsov, and Evan Evans technical discussions. We thank Brian Josey for organizing our initial rate constants table. INCBN IGERT. LJH partially supported by NSF Grant DGE-0549500. LJH and SJK supported by DTRA basic research grant HDTRA-1-09-1-0018.

An open source LabVIEW platform for simulating image series of fluorescent microtubules in gliding assays

Lawrence J. Herskowitz and Steven J. Koch

Introduction

Kinesin family proteins are motor proteins that are able to use chemical energy to translocate on microtubules. They are essential to many cellular processes such as cell reproduction and axonal transport (Goldstein and Yang 2000; Vale and Fletterick 1997; Wittmann, Hyman, and Desai 2001). A technique that has proven very valuable to studying kinesin's physical properties is the gliding motility assay (GMA) (Vale, Reese, and Sheetz 1985). In the GMA, the kinesin tail is fixed to a slide with the motor domains exposed to the solution. A solution of microtubule with buffer and ATP are then added to the flow cell. This allows the microtubules to be propelled by the kinesin motor domains as depicted in Figure 4.1. Often the microtubules are labeled with dye molecules that allow them to be seen through fluorescence microscopy (Greene and Henikoff 2010). GMAs enable the study of many kinetic properties of wild type and mutant kinesins, such as gliding speed, microtubule polarity, and force induction in the kinetic process (Clemmens et al. 2003; Dennis, Howard, and Vogel 1999; Hess et al. 2002a; Hess, Howard, and Vogel 2002b).

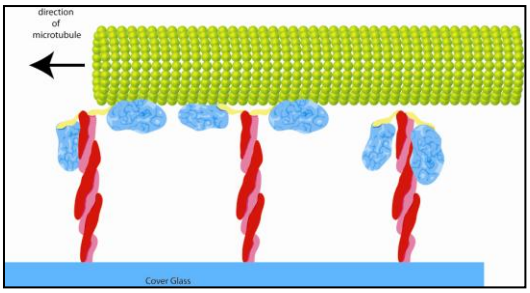


Figure 4.1. Artistic illustration of gliding motility assay. The kinesin are attached to the cover glass so that the motor domains are exposed to the flow cell. When ATP is in the solution, the kinesin will walk along the microtubule which then will be propelled forward.

To extract information from these experiments, it is often imperative to track the moving microtubules through a series of images, obtaining the position versus time of the microtubules. However, manual tracking means going through hundreds of images and recording the position of the microtubule by hand. This is a tedious and exhausting process, though it certainly has proven useful (Howard 1996). The data are also potentially more prone to selection bias and systematic and random errors than automated tracking. Because of this several groups have developed microtubule tracking software (Stuurman 2009; Chisena et al. 2007). To test the effectiveness of these tracking programs, many laboratories will track a stationary object. This however is not ideal since it does not match experimental conditions well. To characterize the systematic and random errors in the tracking algorithms, it is best to test them on a moving object. One way this can be achieved is by moving a fixed microtubule with a piezoelectric stage. However, even in that case it is difficult to completely eliminate drift and other effects. A simulated image series can eliminate those problems and is often ideal for testing tracking algorithms (Cheezum, Walker, and Guilford 2001). For this purpose we developed an application that can generate microtubule images that mimic those captured from a typical gliding motility assay. A simulated image is shown in Figure 4.2A next to an actual gliding motility assay image shown in Figure 4.2B.

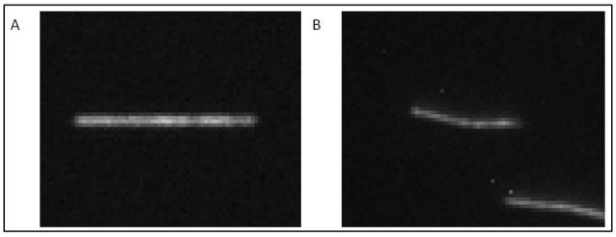


Figure 4.2. Comparison of simulated image (A) with actual image (B). Image A was created using the software written in this paper. Image B is an image of a microtubule from a gliding motility assay. The settings used to create this image can be found at http://kochlab.org/files/Simulating%20Images%20Paper named imitating andys tube.ini

The application was written in National Instruments LABVIEW 7.1. The user interface can be seen in the Figure 4.3 with user inputs on the left and top and outputs on the right and bottom.

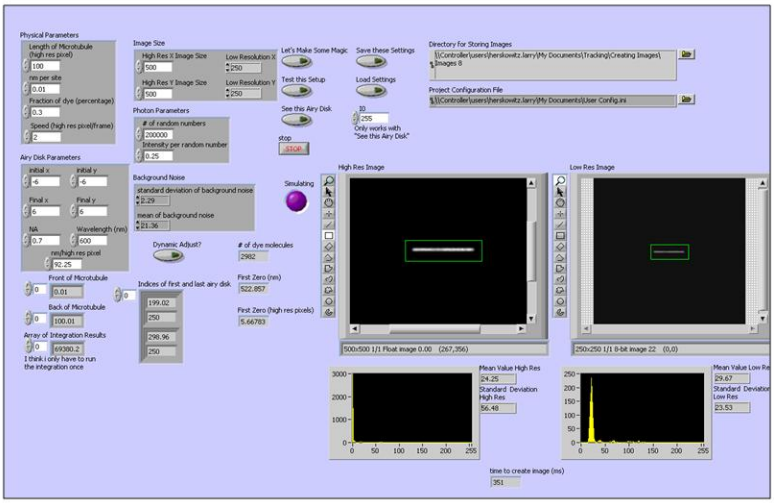


Figure 4.3. User interface of the software. This is an image of the front panel of this software. The top left corner contains the main user inputs that are described in this paper. To their right are the buttons to create a series of images (top left), test this set up to create just one image (second from top on left), below that is the option to see just one airy disc from the "Airy Disk Parameters," after that is the stop button to stop the program. The right column of buttons from top to bottom is to save the settings on the screen and the load settings. The save button saves the settings as an .ini file in the path to the right named "Project Configuration File." The load button will load the settings from that .ini file. Above "Project Configuration File" is the directory location to save the series of images. The two images on this page are the high resolution image (left) and the low resolution image with background noise (left). Below that is the histogram of pixels inside the ROI drawn on the corresponding image. A video tutorial created with camstudio for this software can be found at https://sourceforge.net/project/admin/exploer.php?group_id=312060

Algorithm Overview

Figure 4.4 shows a simplified diagram of the overall algorithm. This paper will go into detail of how each step of the algorithm works. There is a main while loop seen in the diagram (see Reference (Anon 2010) for a LABVIEW video tutorial).

In each iteration a frame is constructed and saved. Before the while loop is entered, three initialization tasks are completed shown as subVIs I, II, III in the figure. (I) First the user must create the trajectory the microtubule will follow in the image series that will be created. This is done in the trajectory subVI. (II) Next using the settings in the “Physical Parameters” control the program randomly sets the locations of the dye molecules along the microtubule length. (III) The third task is to create a prototype Airy Disk from the “Airy Disk Parameters.” Each one of these subVIs is completed only once for each set of images, prior to execution of the while loop that generates each of the individual images.

In the while loop, first the absolute coordinates of the dye molecules are set in the subVI labeled “A” in Figure 4.4. This VI, “Find Coord of Dyes.vi” has inputs of the microtubule trajectory, the index of the start of the microtubule, speed, and length of the microtubule. SubVI A has three outputs: “Coordinates of Dye Locations” (bundle of two double precision numbers), the “Index of Start of Microtubule for next frame” (int32) and the “Index of End of Microtubule” (int32). The dye location coordinates determine the locations of the centers of the airy disks in the simulated images. These are absolute coordinates (relative to the top left of the image) instead of the relative coordinates with respect to the microtubule end that the “Pick sites at random” subVI outputs. The second output, “Index of Start of Microtubule for next frame,” goes into a shift register and replaces the previous iteration’s index for the start of the microtubule. The distance the microtubule moves between frames is calculated from the speed inside of this subVI. This is used to find the index along the trajectory the

microtubule will start in the next frame. The index of the end of the microtubule is calculated using the length of the microtubule value similar to calculating the index for the next starting index.

After subVI A finishes, it passes parameters to subVI B, “Create high res image” including the coordinates of the dye locations, a pointer to an image location and the user inputs “Photon Parameters” and “Image Size.” Inside subVI B are the probability distribution function and the Monte Carlo algorithm to randomly select the photon locations.

Next is subVI C, “Create low res noisy image,” with the inputs of “Image Size,” “Background Noise” and the “High Res Image.” SubVI C resamples the high resolution image to create a lower resolution image that matches experimental resolution, dictated by the user input. If selected, Gaussian noise is added to the background. The output of subVI C is the final image.

SubVI D saves the image in a directory the user has chosen, and named after the frame number which is the iteration number, such as 1.png. When the end of the microtubule index has reached the end of the trajectory array, the while loop stops.

This section will be broken up into four segments to provide details of the above subVIs: Microtubule Construction, Resampling and Noise Addition, Trajectory, and Saving. In each of these segments the necessary user input is highlighted as well as a simplified version of the code.

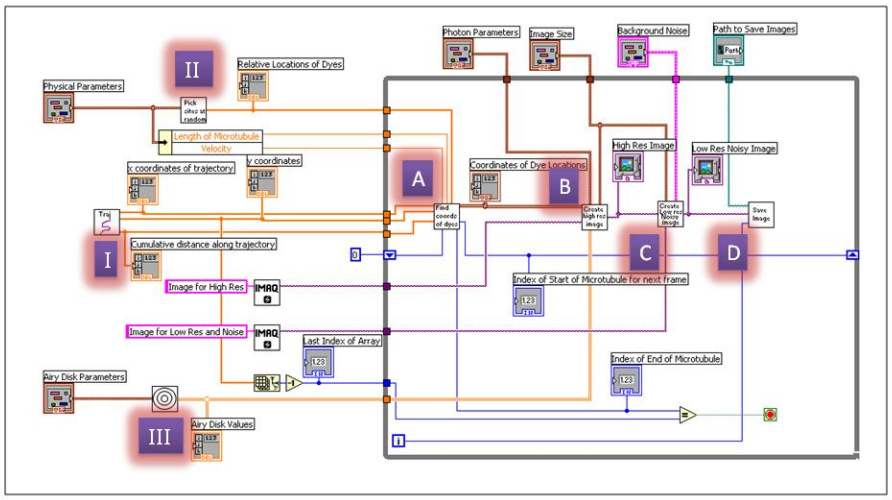


Figure 4.4. Simplified diagram of the image series simulation algorithm. See text for a clear description of the representation of this figure.

Microtubule Construction

User Input:

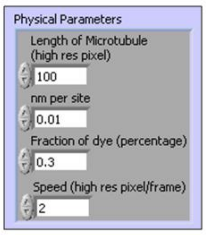


Figure 4.5. Physical Parameters user input. Sets the length of the microtubule, minimum spacing of the dye molecules (pixels per site), probability (out of 1) of dye molecules attaching to each location and the speed of the microtubule.

This section will go into detail of how a single image of a microtubule is constructed. The coordinates for the dyes need to be randomly selected along the length of the microtubule. The user can set the microtubule length via the "Length of Microtubule" control. This control is in units of high resolution pixels. This algorithm produces two images; a high resolution image and a lower resolution image. The final image is the low resolution image. This allows for more accurate numerical estimation of the airy disk probability density function which is implemented in the Monte Carlo algorithm. The "high res pix per site" control dictates how many fluorescent dyes are theoretically possible to be found

in each pixel. For a typical fluorescent tubulin preparation (Anon 2010a), the molecules are attached to surface lysine residues on the alpha/beta tubulin dimer. There is one dimer per 8 nm on a single protofilament. For a 13-protofilament microtubule, there are thus approximately 1.5 dimers per nanometer. There also could be multiple dye molecules on a single dimer, which would further reduce the number of high resolution pixels per dye site. This control will change depending on the users desired nm/pixel ratio.

The first two controls thus dictate the microtubule length and the maximum number of dyes that can be attached. In the example shown in the figure, there can be a maximum of 10,000 dyes. However in practice, fluorescently labeled tubulin is mixed with unlabeled tubulin resulting in partially labeled microtubules after polymerization. To mimic this, the program uses a Monte Carlo method to determine if a given site will be labeled dependent on the fraction of dye control. A uniform random number from 0 to 1 is selected for each site. If that random number is lower than the user-set labeling fraction, the site is deemed as labeled. If not the site is left as unlabeled. This code can be seen in Figure 4.6. There is an indicator on the front panel seen to the right of the dynamic adjust control in Figure 4.3 which specifies the number of dye molecules on the microtubule.

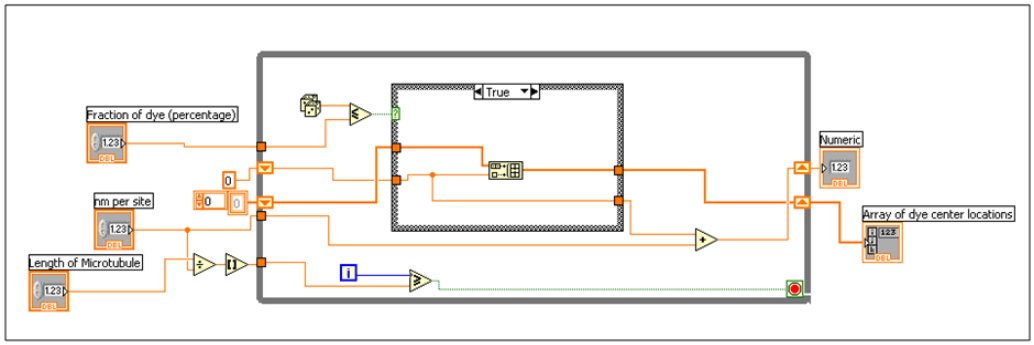


Figure 4.6. Code for determining location of dye molecules. A random number is generated for each possible site. If the random number is lower than probability value input into Fraction then the site is added to the array of dye centers.

If the site is selected to be labeled, an airy disk is centered at that location. An airy disk represents the fluorescent dye in the image plane since the dye emits light that is captured by the circular aperture of the objective (Wolf 1951). Mathematically an airy disk is described with the following equation:

$$I(r) = I_0 \left(\frac{2J_1(kr)}{kr} \right)^2 \tag{4.1}$$

where I_0 is the maximum intensity of the airy disk center, $J_1(kr)$ is the Bessel function of the first kind in circular coordinates, r is the radius in high res pixels, and k is what we have named the prefactor. It can be calculated by

$$k = \frac{2\pi NA}{\lambda} C \tag{4.2}$$

NA is the numerical aperture of the objective, and λ is the wavelength in nanometers. C is the conversion factor from high resolution pixels to nanometers because we represent r in pixels. The first zero for the airy disk can be calculated in nanometers using

$$r_0 = \frac{0.61\lambda}{NA} \tag{4.3}$$

where r_0 is the distance to the first zero (Davidson 2010).

Since the microtubule is narrow compared to the airy disk radius, we treat it as a one-dimensional object. So each airy disk location is recorded as a distance from the end of the microtubule, and these relative distances along the contour remain fixed while the microtubule moves or curves during the simulation.

This method of labeling dyes randomly along the microtubule is supported by prior experimental work. A microtubule in a solution with a low concentration of fluorescent dyes looks speckled. Waterman-Storer and Salmon studied this phenomenon and showed that this is caused by a non-uniform distribution of the fluorescent dyes along the microtubule proto-filament in a purely stochastic process (Waterman-Storer and Salmon 1998).

User Input:

Figure 4.7. Airy Disk Parameters. Airy disk is centered at 0,0. Initial x,y controls set the left, top of the bounding box. Final x,y controls sedt the right, bottom. Limiting the box size increases computation speed. NA sets the numerical aperture value. Wavelength represents the wavelength of the emitted light in nanometers, and the final input is the conversion factor that transforms nm to high res pixels. The final three inputs are used to calculate the prefactor of the airy disk as seen in equation 4.2.

Since the airy disk intensity falls quickly, it is not necessary to calculate the airy disk over large values. The numerical aperture and wavelength determines how quickly the airy disk reaches zero. A higher numerical aperture or shorter wavelength means fewer pixels need to be calculated to get an accurate representation than a higher numerical aperture and longer wavelength. This is illustrated in Figure 4.8. It is obvious that the airy disk in 8A with a radius of 5.67 high res pixels (522.86 nm) needs fewer pixels than 8B whose radius is 19.84

high res pixels (1,830 nm) to approximate the dye molecule. In the user input image above, a square of -6 to 6 for both x and y values was chosen. This means that a 12 pixels x 12 pixel box centered at the airy disk's center was used for each airy disk in the simulation which has the same radius as Figure 4.8A. This is done to ensure a shorter processing time for the software. The radius of the airy disk is calculated in both nanometers and high res pixels. This can be seen on the front panel in Figure 4.3 below the number of dye molecules indicator.

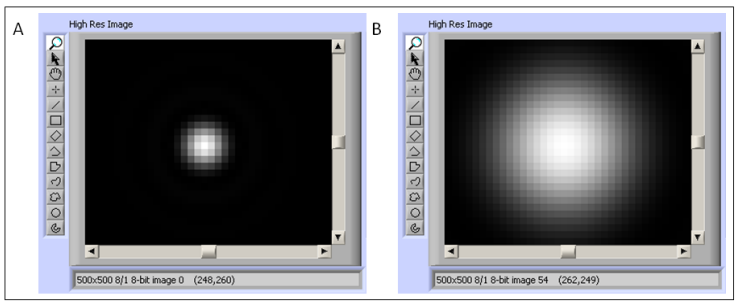


Figure 4.8. Airy disks with differing characteristic sizes. (A) size of 522.86 nm (5.67 high res pixels. (B) size of 1830 nm (19.84 high res pixels). A smaller characteristic size requires a smaller bounding box.

Generating the Total Probability Density Function for Photon Arrival in the Image Plane

User Input:

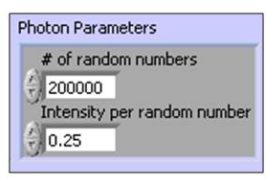


Figure 4.9. Photon Parameters. "# of random numbers" determines how many virtual photons will arrive in the image. "Intensity per random number" determines how much the pixel value will be incremented by one photon.

After all the coordinates of all airy disks have been determined, the functions are added together. This is shown in the equation below.

$$I_T = \sum_{i=0}^{M} I(|\, r_{pixel} - r_{shift} \,|) \tag{4.4}$$

Here I_T is the total intensity of all *M* airy disks added together. $I(r)$ is the intensity profile of a single airy disk. Its center is shifted inside the image by r_{shift} I_T is then normalized by numerical integration to produce the probability density function (PDF) for photon arrival. Each pixel coordinate now represents the probability of a given photon landing there.

Generating Images Based on PDF for Photon Arrival

Photon locations are randomly selected using a Monte Carlo method and the total photon PDF. To minimize the number of random numbers needed for each photon arrival, all values in the two dimensional PDF is flattened into a one-dimensional cumulative probability array. This is seen in Figure 4.10. Because the PDF was normalized, the last element in the cumulative probability array is 1. Each element in the cumulative array has a corresponding pixel coordinate. For a given random number, the index of the cumulative array element closest to that value (without exceeding it) is found by a binary search algorithm. The value of pixel corresponding to that element is increased by an amount set by “intensity per random number.” This method requires only one random number for each photon (Voter 2007). This code is shown in Figure 4.11.

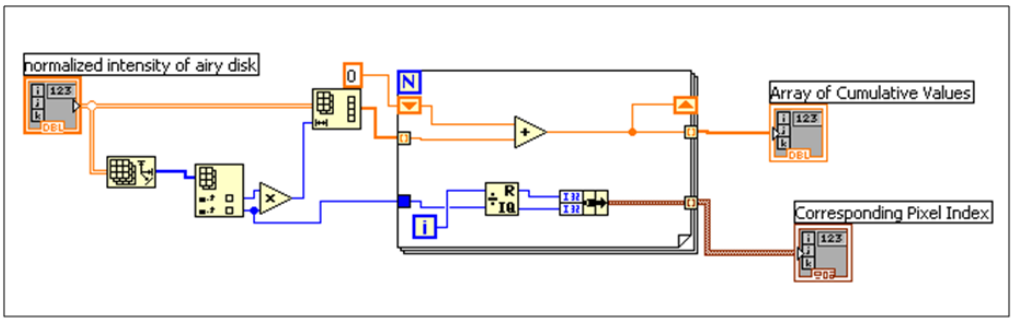


Figure 4.10. Code to create cumulative array used in Monte Carlo algorithm for determining photon location. The normalized intensity of the overlapping airy disks enters this subVI as a 2-d array and is flattened into a 1-d array. This newly formed 1-d array is indexed by the loop where each element is added together. The "quotient and remainder" VI calculates the corresponding pixel coordinates which are bundled together and formed into a 1-d array. Each element in the pixel index array corresponds to an element in the cumulative probability array.

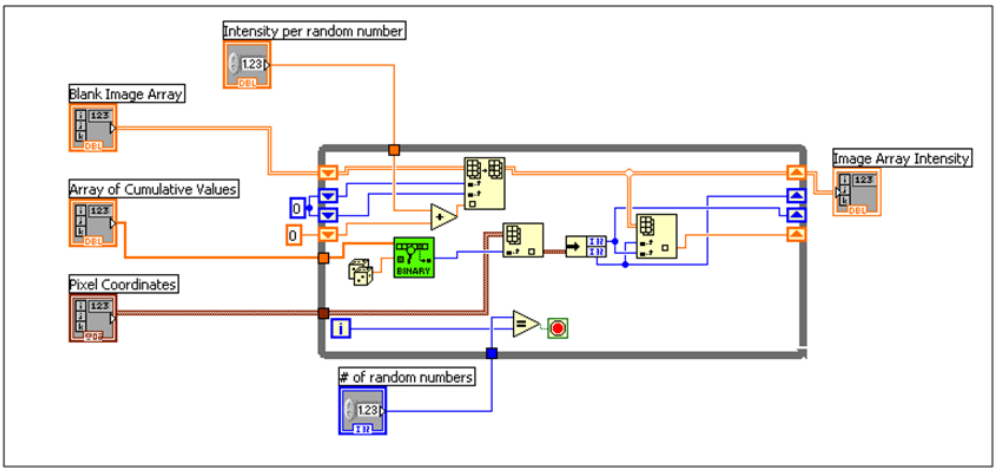


Figure 4.11. Code for the simulating photon arrival in high resolution image. The dice generates a uniformly distributed random number from 0 to 1 and then the green VI (Anon n d) uses a binary search algorithm to find the highest index in the cumulative array whose value is lower than the random number. This index is used to find the corresponding pixel in the Image Array, which is incremented by the amount "Intensity per random number." This while loop continues until it has reached the number of photons specified by the user.

The user chooses how many random numbers, or photons, used for each image. Figure 4.12 represents what an image from this step in the process looks like. The image produced is a floating point image to allow for any number of photons per pixel. The image is next cast to an 8-bit image, with a maximum pixel value of 255. The user is given the option of dynamically adjusting the image so that the maximum pixel value in the floating point image set to 255. If

the user chooses not to do this, the image can mimic either under- or over-exposure. The latter case will show saturated pixels.

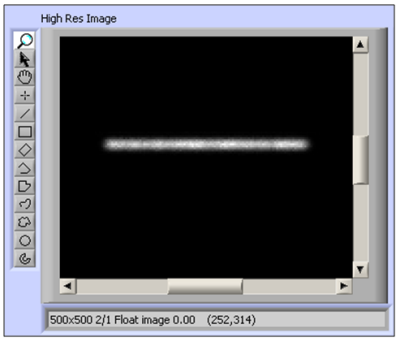


Figure 4.12. Example of the simulated high-resolution microtubule images. Dye-labeling probability of 0.3.

Resampling and Noise Addition

User Input:

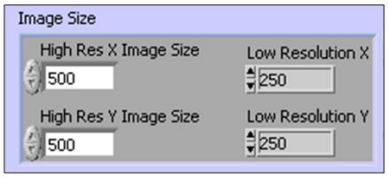


Figure 4.13. Parameters for Image Size. On the left, the width and height (in pixels) of the high-resolution images is set. On the right is the low-resolution image.

After float image is cast into an 8 bit image, the image is then resampled into a lower resolution image to match experimental resolution. The user inputs the x and y resolution of the higher resolution image in “High Res X Image Size” and “High Res Y Image Size.” The user can then control how much the lower resolution image will be in “X Resolution” and “Y Resolution.” This resampling into a lower resolution image is done to allow for a more precise numerical integration to create the probability density function.

User Input:

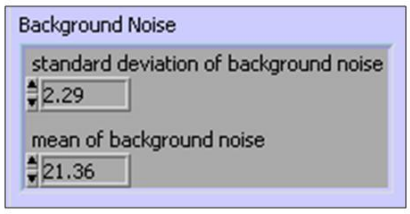


Figure 4.14. Background Noise parameters. The mean and standard deviation (in 8 bit pixel intensity) for Gaussian probability distribution are specified.

Finally, background noise can be added to the image depending on user settings. We add background noise to the 8-bit resampled image. The user chooses the mean value and standard deviation of the background noise Random noise is added to every pixel. The resulting image with its higher resolution counterpart can be seen in Figure 4.15.

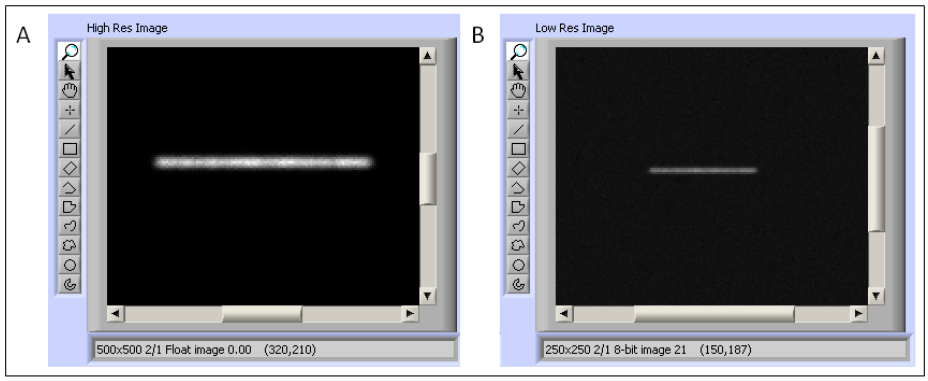


Figure 4.15. Example of image resampling and background noise addition. (A) High resolution image (500 pixels x 500 pixels) without background noise. (B) Image in (A) after resampling (250 pixels x 250 pixels) casting to 8-bit and addition of background noise. See text for details of simulation parameters.

Because there are many parameters it is useful to view test images before creating the entire sequence of images. This option is available to the user along with the additional ability to see what a single airy disk looks like given the current relevant user settings. This allows the user to minimize the box surrounding the airy disk and thus speed up subsequent simulations. Pressing “Test this set up” produces a microtubule while “see this Airy Disk” shows what a single airy disk looks like. The maximum intensity of the airy disk can be

controlled with the “I0.” This only works with “See this Airy Disk” since the sum of all the airy disks is normalized when constructing the microtubule image. These buttons can be seen in Figure 4.3 below the “Let’s Make Some Magic” button.

Trajectory

The ultimate goal of this software is to produce a series of images that mimic microtubule motion in gliding motility assays. This software contains a subVI that allows the user to create a trajectory that the microtubule will follow throughout the frames. The trajectory can be composed of four different base trajectories; a horizontal line, a vertical line, a sloped line, and a circle. The user can manipulate these simpler paths to create elementary or more complicated trajectories as seen in the front panel image of this subVI in Figure 4.16.

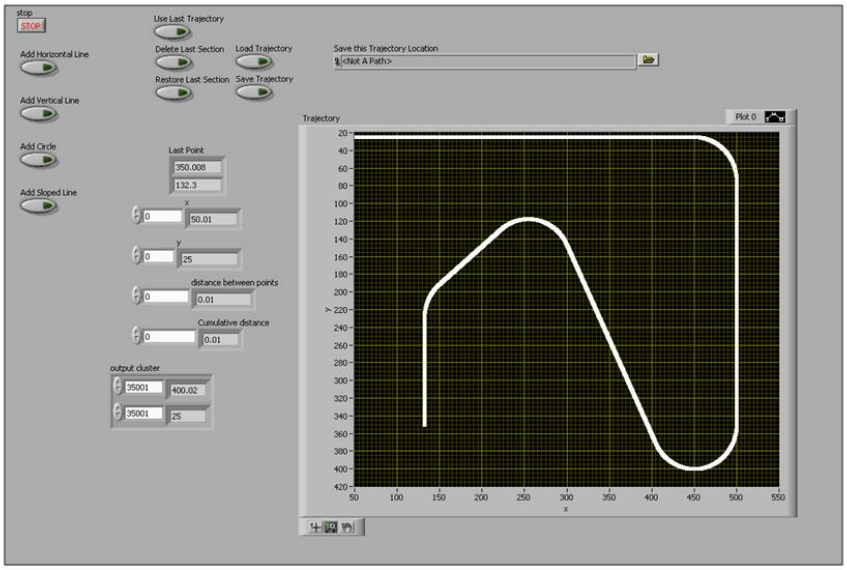


Figure 4.16. Front panel of trajectory subVI. This VI allows the user to design a trajectory the microtubule will follow. The user can add horizontal lines, vertical lines, circles and sloped line to the trajectory. The user can also load previous trajectories and edit them. Trajectories can be saved as files. The current trajectory is shown to the user in the graph seen on the right. See text for further explanation.

The simplest paths are the horizontal and the vertical line. For the horizontal line, the user inputs the starting point coordinate (x,y) and the ending x coordinate. The vertical line is very similar in that it needs the user to input the

starting point coordinate (x,y), but it needs the ending y coordinate instead of x. The sloped line needs both the x and y coordinates for the starting and ending points. It will output the slope of the line for reference. For a circular arc, the user specifies the origin, radius, and angles to start and end the arc. These angles can vary from -2π to 2π.

To make creating more complicated trajectories easier, each one of these paths have an option to automatically link to the existing path. For the horizontal and vertical line this only means that the starting point is set automatically to be the last point of the semi-completed path. For the sloped line, the starting point is set just like the horizontal line, but the slope is also set to the slope of the incoming path. It won't set the slope automatically if the incoming slope is horizontal or vertical. To best attach a circle to the trajectory, the program can set the starting point on the circle so the incoming trajectory has the same slope as the tangential line to the circle. This ensures a smooth transition into the circle. Since there are two points on the circle with the same slope, the user can choose which point to use.

This trajectory subVI also allows the user to load a trajectory previously made and edit that path. The user can also save the completed trajectory as a .dat file in a chosen directory. The main program will automatically save the trajectory in the same directory as the images.

Speed

User Input:

See Figure 4.5

The microtubule will have a fixed speed (see “Updates” section at end of manuscript), defined by the user in units of high res pixel/frame. In the case shown in Figure 4.5 the user has chosen a speed of 2 high res pixels/frame. The amount of resampling will determine the speed in the final, lower resolution images. With the settings shown earlier in Figure 4.5, the tracked speed is 1 low res pixels/frame.

In each image frame, the microtubule dye molecules coordinates will be moved along the trajectory a distance specified by the speed. Since the airy disks are labeled by their distance from the start of the microtubule and not from a location on the image, their relative distances remain unchanged despite curves along the trajectory. The program calculates the distances between adjacent points on the trajectory. Using this array of distances it is able to set the front of the microtubule and quickly search for points that are certain distances away from the start. The code is shown in Figure 4.17.

Finally, when satisfied with the settings and trajectory, the user presses the “Let’s Make Some Magic” button which can be seen in Figure 4.3 to the right of Image Size parameters. The program runs and saves images until it reaches the end of the trajectory. Currently the user cannot stop the simulations early. Pressing the stop button will stop the entire program but will only be responsive after all the images have been created.

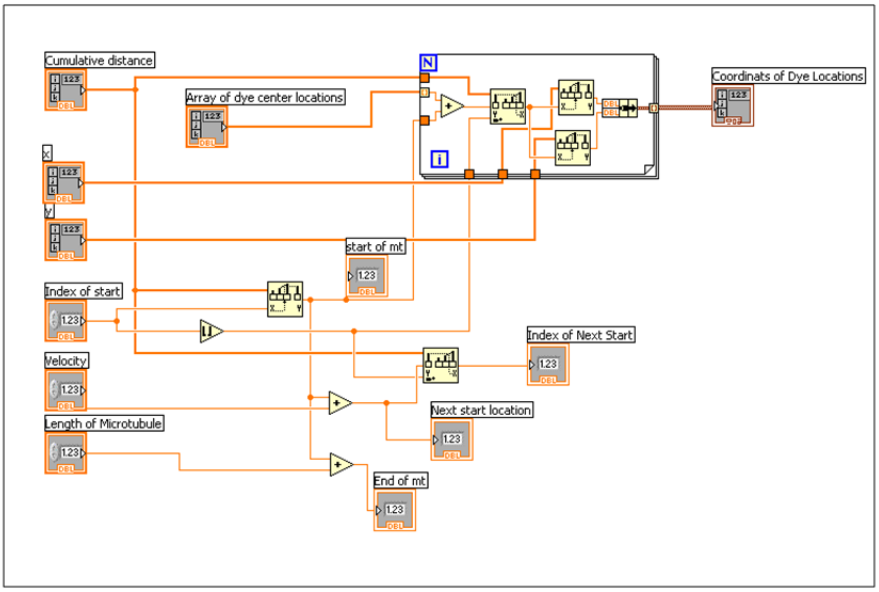


Figure 4.17. Calculation of dye molecule locations. This code calculates the coordinates of the dye locations, the start of the microtubules in the next frame, that point's index along the trajectory, and the end of the microtubule in this frame. The coordinates of the dye locations are calculated by finding a fractional index along the trajectory that corresponds to each dye's distance from the start of the microtubule. This is done by mapping the trajectory onto a straight line using the cumulative distance array (top right). The start of the microtubule and the index to start the microtubule in the next frame are calculated using the specified speed value. The end of the microtubule is found by adding "length of microtubule' to the start value. The program stops creating images when the end of the microtubule reaches the end of the trajectory.

Saving Images and User Settings

User Input:

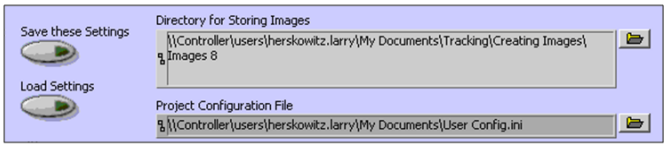


Figure 4.18. Settings and file information. "File Path" specifies the directory in which the images and other information will be saved. "save settings" or "Load Settings" will save / load the current settings to / from the file specified by "Project Configuration File."

This software saves the images in a directory that is specified in "Directory for Storing Images." The images are .pngs and are named after the frame number. The parameters used to create the microtubule as well as the trajectory are saved. The parameters are saved in an .ini file while the trajectory is saved as a .dat file.

This speed of this software is dependent on the user parameters. For example a higher number of photons will slow down the algorithm proportionally.

A larger box around the center of the airy disk will also slow down the simulation. With the settings shown in this paper, the software creates an image in about 350 milliseconds on our Intel Core 2 Duo CPU running Windows XP.

How to Obtain Code and a Video Tutorial

This code is available from sourceforge at https://sourceforge.net/projects/simulatingimage/files/. The VIs used to create this program are available for download in the Simulating Images folder. LabVIEW and the Vision Development Module are required to view and edit the source code. An executable version of this program is also available in Executable Folder found in the Simulating Images directory. A $420 “NI Vision Development Module Run-Time License” is required to run the executable. http://sine.ni.com/nips/cds/view/p/lang/en/nid/207700

A video tutorial created by CamStudio is also available in the Tutorial Folder in the same directory. The .ini file that saved the settings used to create the images found in this paper and the trajectory shown above are located in Settings for Paper folder inside this directory.

Prior Attempts

This report describes our second attempt to create images of microtubules. The first attempt followed more closely to the steps taken in the Cheezum paper (Cheezum, Walker, and Guilford 2001). We found the stochastic method we report here more satisfying as far as mimicking our experimental data. However the other process, based on convolutions of a line or rectangle with an airy disk worked well. We do not describe those methods here, but out work can be seen

in LJH's open notebook, dates 11/12/2009 (http://openwetware.org/wiki/User:TheLarry/Notebook/Larrys_Notebook/2009/11/12) through 11/18/2009 (http://openwetware.org/wiki/User:TheLarry/Notebook/Larrys_Notebook/2009/11/18).

Conclusion

This software can create a series of images of a microtubule moving along a user specified trajectory. This can be used to test tracking software designed for gliding motility assays or other microtubule assays. It is possible to adapt this software to create images of other polymeric protein structures such as f-actin and some cytoskeletal proteins (Danuser and Waterman-Storer 2003; Vallotton et al. 2003). However it is not equipped to handle these yet. The code is freely are available at SourceForge under an MIT license for reuse and adaptation.

Acknowledgements

This research was made possible through DTRA CB Basic Research Program under Grant No. HDTRA1-09-1-008. The authors would also like to acknowledge Anthony Salvagno for creating Figure 4.1. Haiqing Liu for providing the kinesin for the gliding motility assays. Andy Maloney for performing the gliding motility assays especially for providing the image in Figure 2B.

Microtubule Tracking Algorithm for Gliding Motility Assays

Introduction

Kinesin-1, one of the most studied members of the kinesin family, is a molecular motor found in neurons. There are many different types of experiments designed to probe the inner workings of this motor protein including applying a force to the kinesin using an optical tweezer or indirectly measuring the speed of kinesin by calculating the speed of microtubule motion in a gliding motility assay (GMA). In the GMA assay, the kinesin are attached to the cover slide so their catalytic cores are exposed to the solution. Even though the kinesin is immobilized, it still walks along microtubules. This causes the microtubule to move. Speed of the kinesin is measured in this assay by tracking the position of the microtubule and calculating its speed.

Some research groups calculate the speed of the microtubules by hand. They find the pixel location of an end of the microtubule in two different image frames. They then calculate the position change between the frames. These studies have been highly informative, they are not ideal since this process is time consuming and prone to error. To avoid this we have developed software in LabVIEW 7.1 to track the microtubules throughout a series of images and calculate their speeds automatically. There are two parts to this software. The first is the tracking algorithms designed to extract the path of the microtubules throughout the images. The second is the speed calculation algorithm which uses the position data from the tracking algorithm along with the elapsed time that is obtained during image acquisition.

Tracking Algorithm

The core of this algorithm is a pattern matching subVI developed by LabVIEW called IMAQ Match Pattern 2. This subVI searches a given area for matches to a template image, using a cross correlation algorithm. It can find the location of the template within the larger image with subpixel accuracy and allows for rotation of the template. We use a microtubule end as our template. This allows us to track both ends of the microtubule independently. The user can select a template from three options: cropped from an image of their choosing, a simulated image created from the software of the previous chapter, or from alternate software that simulates the image. In the previous chapter the microtubule is simulated using a Monte Carlo technique for photon arrival, which results in a speckled image. The alternate route is to convolute an airy disk with a straight line (Cheezum, Walker, and Guilford 2001). After publishing the previous chapter, we finished implementing the convolution method. We most frequently choose a template from the convolution, since speckle will vary from microtubule to microtubule and inclusion of shot noise in the template only decreases our tracking precision. The differences between the two methods can be seen in Figure 5.1. We do not add noise to the background of the convoluted image. The template created from Figure 5.1 (top) can be seen in Figure 5.2 insert. The pattern matching algorithm has successfully found both ends of the microtubule shown encased in red boxes.

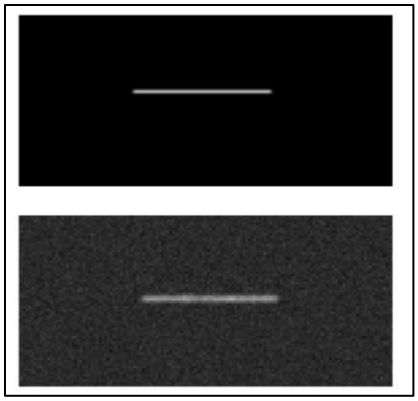


Figure 5.1. Comparison of simulated microtubule images created from convoluting a line with an airy disk (top) and using the speckled method described in the previous chapter (bottom). We found that a template from the top image provided the highest tracking precision.

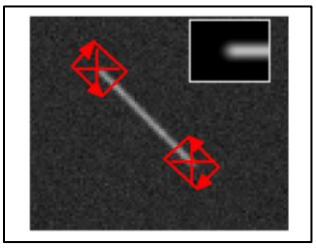


Figure 5.2. An example of the pattern matching algorithm locating both ends of the microtubule (shown in red boxes) with the template shown in the insert. The template is matched to both ends of the microtubule. The red box shows where the template was placed to best match the image. The red crosses show where the algorithm reports the microtubule end.

After choosing the template the next step is for the user to select which microtubule to track from amongst numbered microtubules in an image of the series. An example of these choices can be seen in Figure 5.3. The software picks out the microtubules by loading an image from the movie, and thresholding it automatically. It then finds the white objects in the binary image according to size and elongation. When the user selects a microtubule in the first image, the software finds a rectangle that encloses the microtubule.

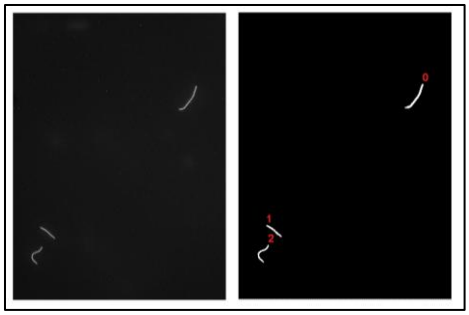


Figure 5.3. Comparison of the original image (left) with the selection image (right). After the original image is converted to a binary image, the algorithm then numbers each microtubule it finds in the image. The user selects the number of the microtubule they wish to track.

This bounding rectangle is used for the first iteration of pattern matching, however for all other iterations of this loop the bounding rectangle is calculated from the previous image. The largest object in the bounding rectangle is outlined and expanded. We choose the largest object since the chances that another unwanted microtubule has a larger area inside the bounding box is low. This expanded search area can be seen in Figure 5.4 (left). In Figure 5.4 (right), the original object is outlined in bright green, and the two ends of the microtubule (red and green dots) and a point on the center of the microtubule (blue dot)are shown. We chose to pattern match in this limited search area to enhance the tracking speed. In order to limit problems from nearby microtubule ends we chose to search an expanded area around the microtubule as opposed to the entire bounding box. We choose to use the outline of the microtubule instead of the bounding rectangle since it is less likely another microtubule gets that close to the tracked microtubule. Using this strategy we encounter fewer errors when two microtubules intersect. The bounding rectangle of the microtubule in the new frame is used for the initial search in the next image as the loop repeats. This process continues until the last image is processed or the user manually ends it.

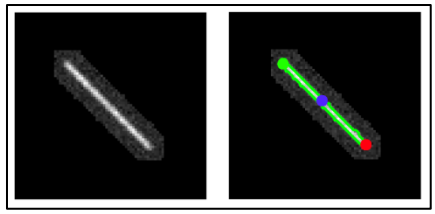


Figure 5.4. Expanded search area (left) and the microtubule outlined in green with the ends, green dot and red dot, and center, blue dot, (right). The microtubule outline (green, right) is expanded to give the algorithm the best chance to match the template to the image, and decreases the possibility of matching the template with another microtubule.

Calculating Speed

Before calculating the speed, the XY position versus time data are smoothed. This is necessary because there is tracking noise and the ends of the microtubule are undergoing Brownian motion. The latter effect is dominant, and without smoothing these transverse fluctuations would systematically increase the speed above the actual longitudinal transport speed. We use a Gaussian weighted sliding window to smooth the x,y data obtained from the tracking algorithm.

For the images we acquire, there is an accompanying time stamp that is accurate to 10's of milliseconds and is used in the speed analysis software. The software also allows for the user to manually input a fixed time per image. Pixel calibration, in nm / pixel, must be entered manually. We calculate the speed by finding $s = \frac{\Delta r}{\Delta t}$ where s is the speed, Δr is calculated by $\Delta r = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$ and Δt is the time between frames. This assumes that the microtubule traveled in a straight line between frames, so it is important for the experimenter to capture images frequently enough for this to be valid.

The speed is calculated between each pair of smoothed XY data points. However, our ultimate goal is to obtain a speed for the microtubule across all image frames. Calculating the mean speed is not sufficient, since the microtubule

may pause or change speeds. We decided to instead estimate the probability density function (PDF) for the speed and use the maximum in the PDF as the speed for that microtubule. We have used a kernel density estimation (KDE) method to estimate the probability density function (PDF) of the speed. We use a Gaussian kernel, which treats each data point as the center of a Gaussian. Each of these Gaussians are added together and normalized, creating an estimate of the speed PDF. KDE can do a good job of extracting different speeds that the microtubule may be traveling. However our current experiments do not look into these differing speeds, and we only use the most likely speed.

Results

All of the images used to verify the effectiveness of the tracking software were created using the software described in the previous chapter. The controls set to create the images can be located at http://openwetware.org/wiki/User:TheLarry/Notebook/Dissertation_Files#INI_Files in the *simulating images settings.ini* file. Unless stated otherwise we set the speed of the simulated microtubules to 1 pixel/frame.

The template used to match the ends of the microtubule in the simulated movies was created by convoluting an airy disk of size 2.8 pixels with a line of zero thickness. The entire microtubule image that was created by the convolution method can be seen in Figure 5.1. The parameters used to make this image can also be found at http://openwetware.org/wiki/User:TheLarry/Notebook/Dissertation_Files#INI_Files in the *Convolution Settings.ini* file.

Finally the tracking software's parameters are also encoded in an .ini file located at the same location called *Tracking settings.ini*. We set the nm/pixel factor to 166.7 nm/pixel which is the calibration value we found using a 60x objective on our microscope. The sec/frame factor was set to .2 sec/frame which is the average capture rate we use when running GMAs. Using these factors the simulated movies have a microtubule speed of 833.5 nm/s.

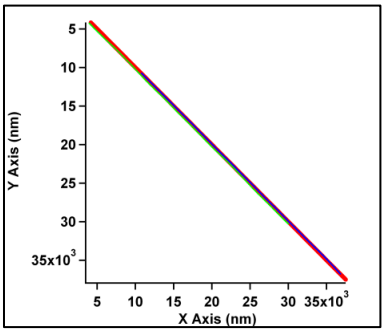


Figure 5.5. Trajectory from the simulated images (red trace) and tracked trajectory for the front (green trace) and back (blue trace) ends of the microtubule. The simulated images followed a trajectory of a diagonal line of slope 1 starting at the top left and ending at the bottom right of the image.

We initially tested the tracking software on a straight diagonal line with constant velocity. The slope of this line was 1 going from the pixel with (x,y) coordinates (25,25) to (250,250). Figure 5.5 shows the actual path taken by the microtubule in red and the tracked path in green (for the front of the microtubule) and blue (back of the microtubule). Clearly the tracking software was able to follow the path well. Figure 5.6 shows the KDE of the PDF for the speed data from this experiment. Data collected from the back and front of the microtubule is seen in blue and red respectively. The PDF has a peak speed of 832.0 nm/s for the front head and 832.7 for the back head. Both are very close to the actual speed. We then ran this experiment over 10 different image series using the same settings

for creating the images and for tracking and recorded a speed of 831.5 $\pm$ 0.9 nm/s (front) and 832.2 $\pm$ 3.2 nm/s.

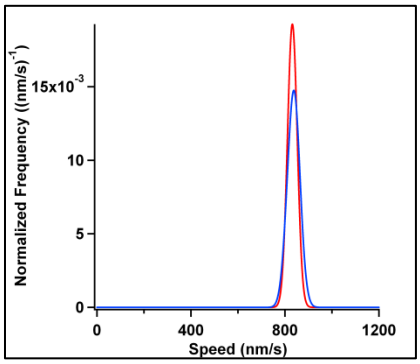


Figure 5.6. KDE of the PDF for the speed data. This KDE shows a single peak for both the front (red) and back (blue) of the microtubule. The highest peaks are located at 832.0 nm/s and 832.7 nm/s for the front and back respectively. The actual speed of the microtubule was 833.5 nm/s.

To test the efficacy of the KDE method for extracting most likely speed, we dded a pausing event into the simulated movie. We used the same trajectory from the previous study, but paused the microtubule for 6 frames at the center of its run. Figure 5.7 shows the speed versus time graph. The blue and red curves are collected from the back and front of the microtubules respectively. The dip in the middle is due to the pausing event. Figure 5.8 shows the KDE produced from the data shown in Figure 5.7. There are two peaks; one associated with the moving speed, and the second from the pausing. The program does not return a value of 0 nm/s for the smaller peak because the smoothing window is wider than the pause duration. The most likely peaks found for the front and back heads respectively were 830.6 nm/s and 829.9 nm/s. The program found a value of 490.7 nm/s and 488.3 nm/s for front and back speeds of the lower peaks. We ran this simulation ten times with the same settings for simulating the images (the microtubule paused for 7 frames) and tracking. The highest peak speed found was 830.9 $\pm$ 0.9 nm/s for the front and 831.0 $\pm$ 1.3 nm/s for the back. The paused

peaks were 490.8 ± 4.7 and 490.7 ± 7.6 for the front and back respectively. If we averaged the speeds instead of using the KDE method we would find a front speed of 789.6 and a back speed of 789.8 nm/s. The KDE method is much less sensitive to pausing than a simple average, as it is closer to the expected value of 833.5 nm/s.

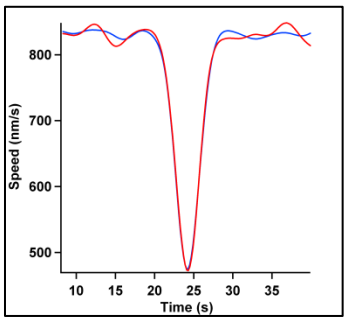


Figure 5.7. Speed data from a tracked microtubule with a pausing event. The microtubule paused for 6 frames between 26.6 s and 27.8 s. Before and after the pausing event the software reported a speed of approximately 830 nm/s. During the pausing event the lowest speed found was 475.2 nm/s. This was not found to be 0 nm/s because the smoothing window was longer than the pausing event and thus incorporated non-zero speeds.

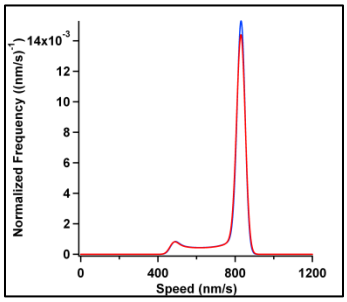


Figure 5.8. KDE from the speed data shown in Figure 5.7. This KDE shows the double peaks we expect since the microtubule traveled at two distinct speeds. The highest peaks were 830.6 nm/s and 829.9 nm/s for the front (red) and back (blue) ends of the microtubule respectively. The lower peaks were 490.7 nm/s and 488.3 nm/s for the front and back respectively. This demonstrates an advantage of the KDE method since it identify when a microtubule is traveling at different speeds.

However there does appear to be remaining a small systematic bias towards lower speeds on the order of a few nanometers per second, but we have not yet attempted to characterize this further. We also have noticed a larger dispersion in the back speeds compared to the front; however we have not studied this yet.

Finally we tested the software by tracking images of microtubules in gliding assay experiments run by Andy Maloney. The data shown here is from one of the

many experiments he ran studying the effects of surface passivation on kinesin velocity. In this experiment he passivated the cover slide with alpha casein. He describes the methods for these experiments in his paper (Maloney, Herskowitz, and Koch 2011) and at his open notebook page http://www.openwetware.org/wiki/User:Andy_Maloney/Kinesin_%26_Microtubule_Page. Seen in Figure 5.9 are screen shots of this experiment. The microtubule is surrounded by its bounding box in green to give it more emphasis in the image and is trailed by the trajectory the software found. The front head trajectory is colored dark green while the back head is light green. This particular microtubule was found going 819.4nm/s for its front head and 818.1 nm/s for its back head. These can be seen as the highest peaks in its PDF seen in Figure 5.10 with the speed from the back end colored in blue and the front in red. This microtubule also apparently traveled at a lower speed, 724.4 nm/s for the front head and 700.2 nm/s for the back head. This shows the power of KDE method to discover bivariate speeds. While we do not know the true speed in these assays, our image tracking software has proven precise and reliable enough to demonstrate differences between assay conditions that produce speed differences on the order of 10 nm/s (Maloney, Herskowitz, and Koch 2011).

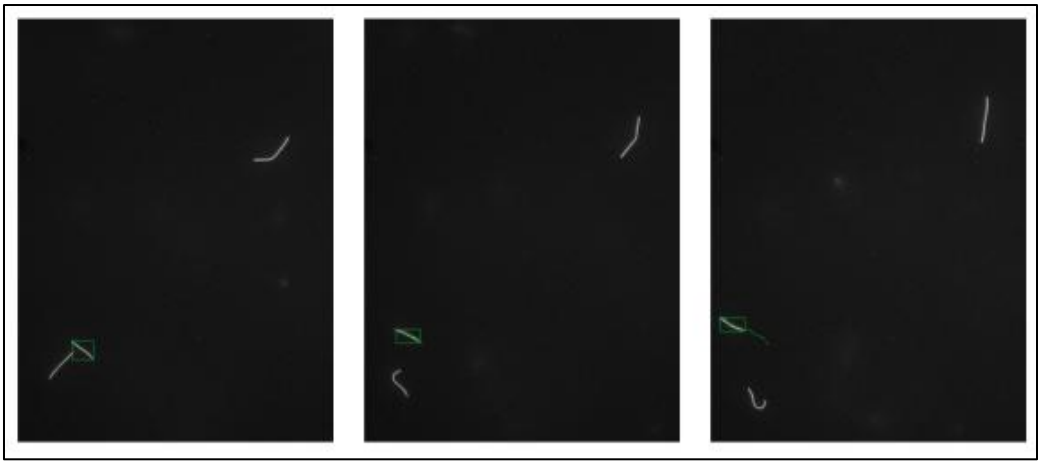


Figure 5.9. Images of a microtubule being tracked from a gliding motility assay. The microtubule being tracked is bounding in a green box. Its trajectory is shown as a dark green line following the front and a light green line following the back of the microtubule.

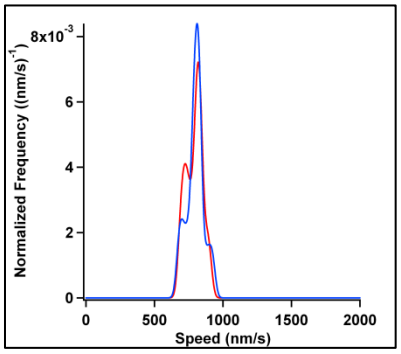


Figure 5.10. KDE found from tracking the microtubule seen in Figure 5.9. Interestingly this KDE shows that the microtubule might have travelled at two different speeds. For our current experiments we only record the highest peak which was 819.4 nm/s and 818.1 nm/s for the front (red) and back (blue) of the microtubule respectively. The lower peaks were located at 724.4 nm/s for the front and 700.2 nm/s for the back of the microtubule.

Conclusion

This tracking software has already proven essential for the gliding motility assays in our lab. However in the future improvements could be made including replacing the smoothing element of the analysis with a hidden Markov method that explicitly models the noise and the motion. One of the greatest improvements on this software which we have implemented and not described here is automation of the entire tracking process.

Once the experimenter captures their images they can tell the software the location of the folder and it will automatically segment all the images and track all

microtubules. It does this by finding all objects in the first image. It will then track each microtubule one at a time. It stops tracking a microtubule when the microtubule leaves the field of view or it crosses over with another microtubule. Once all the microtubules in the first image are tracked, it moves to the second image. The software checks to see if there are any new objects by comparing the bounding box with all the objects in Frame 2 with microtubules already tracked in that frame. If there are any new microtubules, the software will track those as long as possible. Using this method we have automated the tracking process. The speed analysis has also been automated.

Combining these two achievements we have turned a process of tracking and analyzing the hundreds of microtubules found in a single two minute experiment that could take days to finish into one that is done within a couple of hours, requiring only a few minutes of user interaction with the software. This has led to our lab having analyzed terabytes of data on microtubule motion in GMA. This is important as we can more precisely record speed and variance of speed of the microtubules and distinguish more subtle differences in speed that result from change in assay conditions.

We are currently turning this chapter into a paper to be submitted for peer review.

Conclusion

Since each chapter has its own conclusion and future work section it is not necessary to reiterate that information here. Important research can grow out of the work shown here. Moving past a proof of principle for shotgun DNA mapping could have a large impact on cancer studies. I look forward to seeing force added to the kinesin model or probing osmotic pressure effects on processivity. The tracking software allows terabytes of information to be collected from gliding motility assays. The ability to collect that amount of information makes future experiments very exciting. This is one of the joys of science is seeing who picks up these loose ends and what they make of them.

References

Adio, Sarah, Johann Jaud, Bettina Ebbing, Matthias Rief, and Günther Woehlke. 2009. "Dissection of kinesin's processivity." *PLoS ONE* 4:e4612. http://www.ncbi.nlm.nih.gov/pubmed/19242550.

Anon. 2010a. "About Tubulin." http://www.cytoskeleton.com/products/tubulins/abouttub.html.

Anon. n d "Binary Search 1D Array Function for Labview." http://zone.ni.com/devzone/cda/epd/p/id/220.

Anon. 2010b. "Labview Tutorial 1." http://www.youtube.com/watch?v=Em5R_RM8E08.

Arnal, Isabelle, and Richard H Wade. 1998. "Nucleotide-dependent conformations of the kinesin dimer interacting with microtubules." *Structure (London, England : 1993)* 6:33-8. http://www.ncbi.nlm.nih.gov/pubmed/9493265.

Asbury, Charles L. 2005. "Kinesin: world's tiniest biped." *Current opinion in cell biology* 17:89-97. http://www.ncbi.nlm.nih.gov/pubmed/15661524.

Asenjo, Ana B, Yonatan Weinberg, and Hernando Sosa. 2006. "Nucleotide binding and hydrolysis induces a disorder-order transition in the kinesin neck-linker region." *Nature structural & molecular biology* 13:648-54. http://www.ncbi.nlm.nih.gov/pubmed/16783374.

Astumian, R D, and Imre Derényi. 1999. "A chemically reversible Brownian motor: application to kinesin and Ncd." *Biophysical journal* 77:993-1002. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1300390&tool=pmcentrez&rendertype=abstract.

Bell, G I. 1978. "Models for the specific adhesion of cells to cells." *Science (New York, N.Y.)* 200:618-27. http://www.ncbi.nlm.nih.gov/pubmed/347575.

Block, Steven M. 2007. "Kinesin motor mechanics: binding, stepping, tracking, gating, and limping." *Biophysical journal* 92:2986-95. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1852353&tool=pmcentrez&rendertype=abstract.

Block, Steven M, Charles L Asbury, Joshua W Shaevitz, and Matthew J Lang. 2003. "Probing the kinesin reaction cycle with a 2D optical force clamp." *Proceedings of the National Academy of Sciences of the United States of America* 100:2351-6. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=151344&tool=pmcentrez&rendertype=abstract.

Bockelmann, U, Ph Thomen, B Essevaz-Roulet, V Viasnoff, and F Heslot. 2002. "Unzipping DNA with optical tweezers: high sequence sensitivity and force flips." *Biophysical journal* 82:1537-53. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1301953&tool=pmcentrez&rendertype=abstract.

Boeger, Hinrich, Joachim Griesenbeck, and Roger D Kornberg. 2008. "Nucleosome retention and the stochastic nature of promoter chromatin remodeling for transcription." *Cell* 133:716-26. http://www.ncbi.nlm.nih.gov/pubmed/18485878.

Buratowski, Stephen. 2008. "Transcription. Gene expression--where to start?" *Science (New York, N.Y.)* 322:1804-5. http://www.ncbi.nlm.nih.gov/pubmed/19095933.

Bustamante, Carlos, Yann R Chemla, Nancy R Forde, and David Izhaky. 2004. "Mechanical processes in biochemistry." *Annual review of biochemistry* 73:705-48. http://www.ncbi.nlm.nih.gov/pubmed/15189157.

Carter, Nicholas J, and Robert A Cross. 2006. "Kinesin's moonwalk." *Current opinion in cell biology* 18:61-7. http://www.ncbi.nlm.nih.gov/pubmed/16361092.

Carter, Nicholas J, and Robert A Cross. 2005. "Mechanics of the kinesin step." *Nature* 435:308-12. http://www.ncbi.nlm.nih.gov/pubmed/15902249.

Case, Ryan, D W Pierce, N Hom-Booher, Cindy Hart, and Ronald D Vale. 1997. "The directional preference of kinesin motors is specified by an element outside of the motor catalytic domain." *Cell* 90:959-66. http://www.ncbi.nlm.nih.gov/pubmed/9298907.

Cheezum, M K, W F Walker, and W H Guilford. 2001. "Quantitative comparison of algorithms for tracking single fluorescent particles." *Biophysical journal* 81:2378-88. http://www.ncbi.nlm.nih.gov/pubmed/11566807.

Chisena, Ernest N, R Andrew Wall, Jed C Macosko, and George Holzwarth. 2007. "Speckled microtubules improve tracking in motor-protein gliding assays." *Physical biology* 4:10-5. http://www.ncbi.nlm.nih.gov/pubmed/17406081.

Clemmens, John et al. 2003. "Mechanisms of Microtubule Guiding on Microfabricated Kinesin-Coated Surfaces: Chemical and Topographic Surface Patterns." *Langmuir* 19:10967-10974. http://pubs.acs.org/doi/abs/10.1021/la035519y.

Coleman, Michael. 2005. "Axon degeneration mechanisms: commonality amid diversity." *Nature reviews. Neuroscience* 6:889-98. http://www.ncbi.nlm.nih.gov/pubmed/16224497.

Coppin, C M, D W Pierce, L Hsu, and Ronald D Vale. 1997. "The load dependence of kinesin's mechanical cycle." *Proceedings of the National Academy of Sciences of*

the United States of America 94:8539-44. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=23000&tool=pmcentrez&rendertype=abstract.

Core, Leighton J, and John T Lis. 2008. "Transcription regulation through promoter-proximal pausing of RNA polymerase II." *Science (New York, N.Y.)* 319:1791-2. http://www.ncbi.nlm.nih.gov/pubmed/18369138.

Core, Leighton J, Joshua J Waterfall, and John T Lis. 2008. "Nascent RNA sequencing reveals widespread pausing and divergent initiation at human promoters." *Science (New York, N.Y.)* 322:1845-8. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2833333&tool=pmcentrez&rendertype=abstract.

Cross, Robert A. 2004. "The kinetic mechanism of kinesin." *Trends in biochemical sciences* 29:301-9. http://www.ncbi.nlm.nih.gov/pubmed/15276184.

Cross, Robert A et al. 2000. "The conformational cycle of kinesin." *Philosophical Transactions of the Royal Society B: Biological Sciences* 355:459. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1692756.

Cui, Yujia, and Carlos Bustamante. 2000. "Pulling a single chromatin fiber reveals the forces that maintain its higher-order structure." *Proceedings of the National Academy of Sciences of the United States of America* 97:127-32. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=26627&tool=pmcentrez&rendertype=abstract.

Danuser, G, and C M Waterman-Storer. 2003. "Quantitative fluorescent speckle microscopy: where it came from and where it is going." *Journal of microscopy* 211:191-207. http://www.ncbi.nlm.nih.gov/pubmed/12950468.

Davidson, Michael W. 2010. "Numerical Aperture and Image Resolution." http://www.microscopyu.com/tutorials/java/imageformation/airyna/.

De Vos, Kurt J, Andrew J Grierson, Steven Ackerley, and Christopher C J Miller. 2008. "Role of axonal transport in neurodegenerative diseases." *Annual review of neuroscience* 31:151-73. http://www.ncbi.nlm.nih.gov/pubmed/18558852.

Dennis, John R., Jonathon Howard, and Viola Vogel. 1999. "Molecular shuttles: directed motion of microtubules along nanoscale kinesin tracks." *Nanotechnology* 10:232-236. http://iopscience.iop.org/0957-4484/10/3/302.

Derényi, Imre, and T Vicsek. 1996. "The kinesin walk: a dynamic model with elastically coupled heads." *Proceedings of the National Academy of Sciences of the United States of America* 93:6775-9.

http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=39103&tool=pmcentrez&rendertype=abstract.

Duncan, Jason E, and Lawrence S B Goldstein. 2006. "The genetics of axonal transport and axonal transport disorders." *PLoS genetics* 2:e124. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1584265&tool=pmcentrez&rendertype=abstract.

Fan, Dagong, Wenwei Zheng, Ruizheng Hou, Fuli Li, and Zhisong Wang. 2008. "Modeling motility of the kinesin dimer from molecular properties of individual monomers." *Biochemistry* 47:4733-42. http://www.ncbi.nlm.nih.gov/pubmed/18370409.

Farrell, Christopher M, Andrew T Mackey, Lisa M Klumpp, and Susan P Gilbert. 2002. "The role of ATP hydrolysis for kinesin processivity." *The Journal of biological chemistry* 277:17079-87. http://www.ncbi.nlm.nih.gov/pubmed/11864969.

Fisher, Michael E, and Anatoly B Kolomeisky. 2001. "Simple mechanochemistry describes the dynamics of kinesin molecules." *Proceedings of the National Academy of Sciences of the United States of America* 98:7748-53. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=35413&tool=pmcentrez&rendertype=abstract.

Ghosh, Kingshuk, and Ken A Dill. 2006. "Teaching the principles of statistical dynamics." *Physics* 123-133.

Gilbert, Susan P, Michele L Moyer, and Kenneth A Johnson. 1998a. "Alternating site mechanism of the kinesin ATPase." *Biochemistry* 37:792-9. http://www.ncbi.nlm.nih.gov/pubmed/9454568.

Gilbert, Susan P, Michele L Moyer, and Kenneth A Johnson. 1998b. "Pathway of ATP hydrolysis by monomeric and dimeric kinesin." *Biochemistry* 37:800-13. http://www.ncbi.nlm.nih.gov/pubmed/9454569.

Gilbert, Susan P, M R Webb, Kenneth A Johnson, and M. Brune. 1995. "Pathway of processive ATP hydrolysis by kinesin." *Nature* 373:671-6. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1855160&tool=pmcentrez&rendertype=abstract.

Gillespie, Daniel T. 1976. "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions." *Journal of Computational Physics* 22:403-434. http://linkinghub.elsevier.com/retrieve/pii/0021999176900413.

Gillespie, Daniel T, Yang Cao, Kevin R Sanft, and Linda R Petzold. 2009. "The subtle business of model reduction for stochastic chemical kinetics." *The Journal of chemical physics* 130:064103. http://www.ncbi.nlm.nih.gov/pubmed/19222263.

Gindhart, Joseph G. 2006. “Towards an understanding of kinesin-1 dependent transport pathways through the study of protein-protein interactions.” *Briefings in functional genomics & proteomics* 5:74-86. http://www.ncbi.nlm.nih.gov/pubmed/16769683.

Goldstein, Lawrence S B, and Alastair Valentine Philp. 1999. “The road less traveled: emerging principles of kinesin motor utilization.” *Annual review of cell and developmental biology* 15:141-83. http://www.ncbi.nlm.nih.gov/pubmed/10611960.

Goldstein, Lawrence S B, and Zhaohuai Yang. 2000. “Microtubule-based transport systems in neurons: the roles of kinesins and dyneins.” *Annual review of neuroscience* 23:39-71. http://www.ncbi.nlm.nih.gov/pubmed/10845058.

Greene, Liz, and Steve Henikoff. 2010. “Kinesin Home Page.” http://www.cellbio.duke.edu/kinesin/.

Grinstead, Charles M., and J. L. Snell. 1997. “Introduction to Probability.” Pp. 405-452 in. American Mathematical Society http://www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/book.html.

Guydosh, Nicholas R, and Steven M Block. 2006. “Backsteps induced by nucleotide analogs suggest the front head of kinesin is gated by strain.” *Proceedings of the National Academy of Sciences of the United States of America* 103:8054-9. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1472428&tool=pmcentrez&rendertype=abstract.

Guydosh, Nicholas R, and Steven M Block. 2009. “Direct observation of the binding state of the kinesin head to the microtubule.” *Nature* 461:125-8. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2859689&tool=pmcentrez&rendertype=abstract.

Hackney, David D. 1988. “Kinesin ATPase: rate-limiting ADP release.” *Proceedings of the National Academy of Sciences of the United States of America* 85:6314-8. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=281960&tool=pmcentrez&rendertype=abstract.

Hackney, David D. 1996. “The kinetic cycles of myosin, kinesin, and dynein.” *Annual review of physiology* 58:731-50. http://www.ncbi.nlm.nih.gov/pubmed/8815818.

Hahlen, Katrin et al. 2006. “Feedback of the kinesin-1 neck-linker position on the catalytic site.” *The Journal of biological chemistry* 281:18868-77. http://www.ncbi.nlm.nih.gov/pubmed/16682419.

Hall, Michael A et al. 2009. “High-resolution dynamic mapping of histone-DNA interactions in a nucleosome.” *Nature structural & molecular biology* 16:124-9.

http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2635915&tool=pmcentrez&rendertype=abstract.

Hancock, William O, and Jonathon Howard. 1999. "Kinesin's processivity results from mechanical and chemical coordination between the ATP hydrolysis cycles of the two motor domains." *Proceedings of the National Academy of Sciences of the United States of America* 96:13147-52. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=23915&tool=pmcentrez&rendertype=abstract.

Hariharan, Venkatesh, and William O Hancock. 2009. "Insights into the Mechanical Properties of the Kinesin Neck Linker Domain from Sequence Analysis and Molecular Dynamics Simulations." *Cellular and Molecular Bioengineering* 2:177-189. http://www.springerlink.com/index/10.1007/s12195-009-0059-5.

He, Yiping, Bert Vogelstein, Victor E Velculescu, Nickolas Papadopoulos, and Kenneth W Kinzler. 2008. "The antisense transcriptomes of human cells." *Science (New York, N.Y.)* 322:1855-7. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2824178&tool=pmcentrez&rendertype=abstract.

Hess, Henry et al. 2002a. "Ratchet patterns sort molecular shuttles." *Applied Physics A: Materials Science & Processing* 75:309-313. http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1007/s003390201339.

Hess, Henry, Jonathon Howard, and Viola Vogel. 2002b. "A Piconewton Forcemeter Assembled from Microtubules and Kinesins." *Nano Letters* 2:1113-1115. http://pubs.acs.org/doi/abs/10.1021/nl025724i.

Hirokawa, Nobutaka. 1998. "Kinesin and dynein superfamily proteins and the mechanism of organelle transport." *Science (New York, N.Y.)* 279:519-26. http://www.ncbi.nlm.nih.gov/pubmed/9438838.

Hirokawa, Nobutaka et al. 1991. "Kinesin associates with anterogradely transported membranous organelles in vivo." *The Journal of cell biology* 114:295-302. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2289077&tool=pmcentrez&rendertype=abstract.

Hirokawa, Nobutaka, and Reiko Takemura. 2005. "Molecular motors and mechanisms of directional transport in neurons." *Nature reviews. Neuroscience* 6:201-14. http://www.ncbi.nlm.nih.gov/pubmed/15711600.

Howard, Jonathon. 1996. "The movement of kinesin along microtubules." *Annual review of physiology* 58:703-29. http://www.ncbi.nlm.nih.gov/pubmed/8815816.

Hua, W, E C Young, M L Fleming, and Jeff Gelles. 1997. "Coupling of kinesin steps to ATP hydrolysis." *Nature* 388:390-3. http://www.ncbi.nlm.nih.gov/pubmed/9237758.

Hunt, A J, F Gittes, and Jonathon Howard. 1994. "The force exerted by a single kinesin molecule against a viscous load." *Biophysical journal* 67:766-81. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1225420&tool=pmcentrez&rendertype=abstract.

Hurd, D D, and W M Saxton. 1996. "Kinesin mutations cause motor neuron disease phenotypes by disrupting fast axonal transport in Drosophila." *Genetics* 144:1075-85. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1207603&tool=pmcentrez&rendertype=abstract.

Hyeon, Changbong, Stefan Klumpp, and José N Onuchic. 2009. "Kinesin's backsteps under mechanical load." *Physical chemistry chemical physics : PCCP* 11:4899-910. http://www.ncbi.nlm.nih.gov/pubmed/19506765.

Hänggi, Peter, and Michal Borkovec. 1990. "Reaction-rate theory: fifty years after Kramers." *Reviews of Modern Physics* 62:251-341. http://link.aps.org/doi/10.1103/RevModPhys.62.251.

Imafuku, Yasuhiro, Neil Thomas, and Katsuhisa Tawada. 2009. "Hopping and stalling of processive molecular motors." *Journal of theoretical biology* 261:43-9. http://www.ncbi.nlm.nih.gov/pubmed/19627996.

Kanada, Ryo, and Kazuo Sasaki. 2003. "Theoretical model for motility and processivity of two-headed molecular motors." *Physical review. E, Statistical, nonlinear, and soft matter physics* 67:061917. http://www.ncbi.nlm.nih.gov/pubmed/16241271.

Kaseda, Kuniyoshi, Hideo Higuchi, and Keiko Hirose. 2003. "Alternate fast and slow stepping of a heterodimeric kinesin molecule." *Nature cell biology* 5:1079-82. http://www.ncbi.nlm.nih.gov/pubmed/14634664.

Kawaguchi, Kenji, and Shin'ichi Ishiwata. 2001. "Nucleotide-dependent single- to double-headed binding of kinesin." *Science (New York, N.Y.)* 291:667-9. http://www.ncbi.nlm.nih.gov/pubmed/11158681.

Kawaguchi, Kenji, Sotaro Uemura, and Shin'ichi Ishiwata. 2003. "Equilibrium and transition between single- and double-headed binding of kinesin as revealed by single-molecule mechanics." *Biophysical journal* 84:1103-13. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1302687&tool=pmcentrez&rendertype=abstract.

Khorasanizadeh, Sepideh. 2004. "The nucleosome: from genomic organization to genomic regulation." *Cell* 116:259-72. http://www.ncbi.nlm.nih.gov/pubmed/14744436.

Kidd, Jeffrey M et al. 2008. “Mapping and sequencing of structural variation from eight human genomes.” *Nature* 453:56-64. http://www.ncbi.nlm.nih.gov/pubmed/18451855.

Kikkawa, Masahide. 2008. “The role of microtubules in processive kinesin movement.” *Trends in cell biology* 18:128-35. http://www.ncbi.nlm.nih.gov/pubmed/18280159.

Kikkawa, Masahide et al. 2001. “Switch-based mechanism of kinesin motors.” *Nature* 411:439-45. http://www.ncbi.nlm.nih.gov/pubmed/11373668.

Koch, Steven J, Alla Shundrovsky, Benjamin C Jantzen, and Michelle D Wang. 2002. “Probing protein-DNA interactions by unzipping a single DNA double helix.” *Biophysical journal* 83:1098-105. http://www.ncbi.nlm.nih.gov/pubmed/12124289.

Kolomeisky, Anatoly B, and Michael E Fisher. 1999. “The force exerted by a molecular motor.” *Proceedings of the National Academy of Sciences of the United States of America* 96:6597-602. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=21960&tool=pmcentrez&rendertype=abstract.

Kordeli, Ekaterini, J Davis, B Trapp, and V Bennett. 1990. “An isoform of ankyrin is localized at nodes of Ranvier in myelinated axons of central and peripheral nerves.” *The Journal of cell biology* 110:1341-52. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2116078&tool=pmcentrez&rendertype=abstract.

Kornberg, Roger D. 1977. “Structure of chromatin.” *Annual review of biochemistry* 46:931-54. http://www.ncbi.nlm.nih.gov/pubmed/332067.

Liao, Jung-Chi, James A Spudich, David Parker, and Scott L Delp. 2007. “Extending the absorbing boundary method to fit dwell-time distributions of molecular motors with complex kinetic pathways.” *Proceedings of the National Academy of Sciences of the United States of America* 104:3171-6. http://www.ncbi.nlm.nih.gov/pubmed/17360624.

Liu, Ming S, B D Todd, and Richard J Sadus. 2005. “Cooperativity in the motor activities of the ATP-fueled molecular motors.” *Biochimica et biophysica acta* 1752:111-23. http://www.ncbi.nlm.nih.gov/pubmed/16140597.

Ma, Y Z, and E W Taylor. 1997a. “Interacting head mechanism of microtubule-kinesin ATPase.” *The Journal of biological chemistry* 272:724-30. http://www.ncbi.nlm.nih.gov/pubmed/8995356.

Ma, Y Z, and E W Taylor. 1997b. “Kinetic mechanism of a monomeric kinesin construct.” *The Journal of biological chemistry* 272:717-23. http://www.ncbi.nlm.nih.gov/pubmed/8995355.

Ma, Y Z, and E W Taylor. 1995. “Kinetic mechanism of kinesin motor domain.” *Biochemistry* 34:13233-41. http://www.ncbi.nlm.nih.gov/pubmed/7548087.

Maloney, R Andy, Lawrence J Herskowitz, and Steven J Koch. 2011. “Effects of Surface Passivation on Gliding Motility Assays.”

Margaritis, Thanasis, and Frank C P Holstege. 2008. “Poised RNA polymerase II gives pause for thought.” *Cell* 133:581-4. http://www.ncbi.nlm.nih.gov/pubmed/18485867.

Martens, J. 2003. “Recent advances in understanding chromatin remodeling by Swi/Snf complexes.” *Current Opinion in Genetics & Development* 13:136-142. http://linkinghub.elsevier.com/retrieve/pii/S0959437X03000224.

Martin, M A, D D Hurd, and W M Saxton. 1999. “Kinesins in the nervous system.” *Cellular and molecular life sciences : CMLS* 56:200-16. http://www.ncbi.nlm.nih.gov/pubmed/11212348.

Miyazono, Yuya, Masahito Hayashi, Peter Karagiannis, Yoshie Harada, and Hisashi Tadakuma. 2010. “Strain through the neck linker ensures processive runs: a DNA-kinesin hybrid nanomachine study.” *The EMBO journal* 29:93-106. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2775897&tool=pmcentrez&rendertype=abstract.

Morfini, Gerardo A et al. 2009. “Axonal transport defects in neurodegenerative diseases.” *The Journal of neuroscience : the official journal of the Society for Neuroscience* 29:12776-86. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2801051&tool=pmcentrez&rendertype=abstract.

Muqit, Miratul M K, and Mel B Feany. 2002. “Modelling neurodegenerative diseases in Drosophila: a fruitful approach?” *Nature reviews. Neuroscience* 3:237-43. http://www.ncbi.nlm.nih.gov/pubmed/11994755.

Muresan, V. 2000. “One axon, many kinesins: What’s the logic?” *Journal of neurocytology* 29:799-818. http://www.ncbi.nlm.nih.gov/pubmed/11466472.

Muse, Ginger W et al. 2007. “RNA polymerase is poised for activation across the genome.” *Nature genetics* 39:1507-11. http://www.ncbi.nlm.nih.gov/pubmed/17994021.

Muthukrishnan, Gayatri, Yangrong Zhang, Shankar Shastry, and William O Hancock. 2009. “The processivity of kinesin-2 motors suggests diminished front-head gating.” *Current biology : CB* 19:442-7. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2730044&tool=pmcentrez&rendertype=abstract.

Nakata, Takao, and Nobutaka Hirokawa. 2003. “Microtubules provide directional cues for polarized axonal transport through interaction with kinesin motor head.” *The Journal of cell biology* 162:1045-55. http://www.ncbi.nlm.nih.gov/pubmed/12975348.

O’Grady, Patrick M. 2010. “Whither Drosophila?” *Genetics* 185:703-5. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2881149&tool=pmcentrez&rendertype=abstract.

O’Grady, Patrick M, and Therese A Markow. 2009. “Phylogenetic taxonomy in Drosophila.” *Fly* 3:10-4. http://www.ncbi.nlm.nih.gov/pubmed/19556883.

Peskin, Charles S., and George Oster. 1995. “Coordinated hydrolysis explains the mechanical behavior of kinesin.” *Biophysical Journal* 68:202S. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1281917.

Rice, Sarah et al. 1999. “A structural change in the kinesin motor protein that drives motility.” *Nature* 402:778-84. http://www.ncbi.nlm.nih.gov/pubmed/10617199.

Rice, Sarah et al. 1999. “A structural change in the kinesin motor protein that drives motility.” *Nature* 402:778-84. http://www.ncbi.nlm.nih.gov/pubmed/10617199.

Rosenfeld, Steven S, Polly M Fordyce, Geraldine M Jefferson, Peter H King, and Steven M Block. 2003. “Stepping and stretching. How kinesin uses internal strain to walk processively.” *The Journal of biological chemistry* 278:18550-6. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1533991&tool=pmcentrez&rendertype=abstract.

Rosenfeld, Steven S, Jun Xing, Geraldine M Jefferson, Herbert C Cheung, and Peter H King. 2002. “Measuring kinesin’s first step.” *The Journal of biological chemistry* 277:36731-9. http://www.ncbi.nlm.nih.gov/pubmed/12122000.

Roy, Subhojit et al. 2000. “Neurofilaments are transported rapidly but intermittently in axons: implications for slow axonal transport.” *The Journal of neuroscience : the official journal of the Society for Neuroscience* 20:6849-61. http://www.ncbi.nlm.nih.gov/pubmed/10995829.

SantaLucia, J. 1998. “A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics.” *Proceedings of the National Academy of Sciences of the United States of America* 95:1460-5. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=19045&tool=pmcentrez&rendertype=abstract.

Schief, William R, and Jonathon Howard. 2001. “Conformational changes during kinesin motility.” *Current opinion in cell biology* 13:19-28. http://www.ncbi.nlm.nih.gov/pubmed/11163129.

Schnapp, Bruce J, and Thomas S Reese. 1989. “Dynein is the motor for retrograde axonal transport of organelles.” *Proceedings of the National Academy of Sciences of the United States of America* 86:1548-52. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=286735&tool=pmcentrez&rendertype=abstract.

Schnitzer, M J, and Steven M Block. 1997. “Kinesin hydrolyses one ATP per 8-nm step.” *Nature* 388:386-90. http://www.ncbi.nlm.nih.gov/pubmed/9237757.

Schnitzer, M J, Steven M Block, and K Visscher. 2000. “Force production by single kinesin motors.” *Nature cell biology* 2:718-23. http://www.ncbi.nlm.nih.gov/pubmed/11025662.

Shao, Qiang, and Yi Qin Gao. 2007. “Asymmetry in kinesin walking.” *Biochemistry* 46:9098-106. http://www.ncbi.nlm.nih.gov/pubmed/17630771.

Shao, Qiang, and Yi Qin Gao. 2006. “On the hand-over-hand mechanism of kinesin.” *Proceedings of the National Academy of Sciences of the United States of America* 103:8072-7. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1472431&tool=pmcentrez&rendertype=abstract.

Shastry, Shankar, and William O Hancock. 2010. “Neck linker length determines the degree of processivity in kinesin-1 and kinesin-2 motors.” *Current biology : CB* 20:939-43. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2882250&tool=pmcentrez&rendertype=abstract.

Shundrovsky, Alla, Corey L Smith, John T Lis, Craig L Peterson, and Michelle D Wang. 2006. “Probing SWI/SNF remodeling of the nucleosome by unzipping single DNA molecules.” *Nature structural & molecular biology* 13:549-54. http://www.ncbi.nlm.nih.gov/pubmed/16732285.

Sikes, Martha, Ann Beyer, and Yvonne Osheim. 2002. “EM visualization of Pol II genes in Drosophila : most genes terminate without prior 3′ end cleavage of nascent transcripts.” *Chromosoma* 111:1-12. http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1007/s00412-002-0183-7.

Silverman, B.W. 1986. *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.

Sindelar, Charles V et al. 2002. “Two conformations in the human kinesin power stroke defined by X-ray crystallography and EPR spectroscopy.” *Nature structural biology* 9:844-8. http://www.ncbi.nlm.nih.gov/pubmed/12368902.

Stewart, William J. 1994. *Introduction to the Numerical Solution of Markov Chains*. Princeton: Princeton University Press.

Stock, Gerhard, Kingshuk Ghosh, and Ken A Dill. 2008. "Maximum Caliber: a variational approach applied to two-state dynamics." *The Journal of chemical physics* 128:194102. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2671656&tool=pmcentrez&rendertype=abstract.

Stokin, Gorazd B et al. 2005. "Axonopathy and transport deficits early in the pathogenesis of Alzheimer's disease." *Science (New York, N.Y.)* 307:1282-8. http://www.ncbi.nlm.nih.gov/pubmed/15731448.

Stuurman, Nico. 2009. "MTrack2." http://valelab.ucsf.edu/~nico/IJplugins/MTrack2.html.

Sudarsanam, P. 2000. "The Swi/Snf family nucleosome-remodeling complexes and transcriptional control." *Trends in Genetics* 16:345-351.

Svoboda, K, C F Schmidt, Bruce J Schnapp, and Steven M Block. 1993. "Direct observation of kinesin stepping by optical trapping interferometry." *Nature* 365:721-7. http://www.ncbi.nlm.nih.gov/pubmed/8413650.

Thomas, Neil, Yasuhiro Imafuku, Tsutomu Kamiya, and Katsuhisa Tawada. 2002. "Kinesin: a molecular motor with a spring in its step." *Proceedings. Biological sciences / The Royal Society* 269:2363-71. http://www.ncbi.nlm.nih.gov/pubmed/12495505.

Thorn, K S, J A Ubersax, and Ronald D Vale. 2000. "Engineering the processive run length of the kinesin motor." *The Journal of cell biology* 151:1093-100. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2174356&tool=pmcentrez&rendertype=abstract.

Tomishige, Michio, Nico Stuurman, and Ronald D Vale. 2006. "Single-molecule observations of neck linker conformational changes in the kinesin motor protein." *Nature structural & molecular biology* 13:887-94. http://www.ncbi.nlm.nih.gov/pubmed/17013387.

Vale, Ronald D, Ryan Case, Elena P Sablin, Cindy Hart, and Robert Fletterick. 2000. "Searching for kinesin's mechanical amplifier." *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 355:449-57. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1692751&tool=pmcentrez&rendertype=abstract.

Vale, Ronald D, and Robert Fletterick. 1997. "The design plan of kinesin motors." *Annual review of cell and developmental biology* 13:745-77. http://www.ncbi.nlm.nih.gov/pubmed/9442886.

Vale, Ronald D et al. 1996. “Direct observation of single kinesin molecules moving along microtubules.” *Nature* 380:451-3. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2852185&tool=pmcentrez&rendertype=abstract.

Vale, Ronald D, and R A Milligan. 2000a. “The way things move: looking under the hood of molecular motor proteins.” *Science (New York, N.Y.)* 288:88-95. http://www.ncbi.nlm.nih.gov/pubmed/10753125.

Vale, Ronald D, and R A Milligan. 2000b. “The way things move: looking under the hood of molecular motor proteins.” *Science (New York, N.Y.)* 288:88-95. http://www.ncbi.nlm.nih.gov/pubmed/10753125.

Vale, Ronald D, Thomas S Reese, and M P Sheetz. 1985. “Identification of a novel force-generating protein, kinesin, involved in microtubule-based motility.” *Cell* 42:39-50. http://www.ncbi.nlm.nih.gov/pubmed/3926325.

Vallotton, P, A Ponti, C M Waterman-Storer, E D Salmon, and G Danuser. 2003. “Recovery, visualization, and analysis of actin and tubulin polymer flow in live cells: a fluorescent speckle microscopy study.” *Biophysical journal* 85:1289-306. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1303246&tool=pmcentrez&rendertype=abstract.

Van Leeuwen, Rudolphus. 2006. “Protein folding and translocation: single-molecule investigations.” http://www.narcis.info/publication/RecordID/oai:openaccess.leidenuniv.nl:1887%2F4991.

Verbrugge, Sander, Zdenek Lansky, and Erwin J G Peterman. 2009. “Kinesin’s step dissected with single-motor FRET.” *Proceedings of the National Academy of Sciences of the United States of America* 106:17741-6. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2764935&tool=pmcentrez&rendertype=abstract.

Verbrugge, Sander, Bettina Lechner, Günther Woehlke, and Erwin J G Peterman. 2009. “Alternating-site mechanism of kinesin-1 characterized by single-molecule FRET using fluorescent ATP analogues.” *Biophysical journal* 97:173-82. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2711353&tool=pmcentrez&rendertype=abstract.

Voter, A.F. 2007. “Introduction to the kinetic Monte Carlo method.” P. 1568–2609 in *Radiation Effects in Solids*, vol. 235. Citeseer http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.125.3560&rep=rep1&type=pdf.

Wagenbach, M, D. L. Coy, and Jonathon Howard. 1999. "Kinesin takes one 8-nm step for each ATP that it hydrolyzes." *The Journal of biological chemistry* 274:3667-71. http://www.ncbi.nlm.nih.gov/pubmed/9920916.

Wang, Lei, Chung-liang Ho, Dongming Sun, Ronald K. H. Liem, and Anthony Brown. 2000. "Rapid movement of axonal neurofilaments interrupted by prolonged pauses." *Nature cell biology* 2:137-41. http://www.ncbi.nlm.nih.gov/pubmed/10707083.

Wang, Michelle D, Hong Yin, Robert Landick, Jeff Gelles, and Steven M Block. 1997. "Stretching DNA with optical tweezers." *Biophysical journal* 72:1335-46. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1184516&tool=pmcentrez&rendertype=abstract.

Wang, Zhisong, Min Feng, Wenwei Zheng, and Dagong Fan. 2007. "Kinesin is an evolutionarily fine-tuned molecular ratchet-and-pawl device of decisively locked direction." *Biophysical journal* 93:3363-72. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2072077&tool=pmcentrez&rendertype=abstract.

Waterman-Storer, C M, and E D Salmon. 1998. "How microtubules get fluorescent speckles." *Biophysical journal* 75:2059-69. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1299878&tool=pmcentrez&rendertype=abstract.

Wittmann, T, A Hyman, and A Desai. 2001. "The spindle: a dynamic assembly of microtubules and motors." *Nature cell biology* 3:E28-34. http://www.ncbi.nlm.nih.gov/pubmed/11146647.

Wolf, E. 1951. "The diffraction theory of aberrations." *Reports on Progress in Physics* 14:95-120. http://stacks.iop.org/0034-4885/14/i=1/a=304?key=crossref.514517e37949fc6df7afa11aad202a61.

Workman, J L, and R E Kingston. 1998. "Alteration of nucleosome structure as a mechanism of transcriptional regulation." *Annual review of biochemistry* 67:545-79. http://www.ncbi.nlm.nih.gov/pubmed/9759497.

Wu, Yanjue, and Yuan Luo. 2005. "Transgenic C. elegans as a model in Alzheimer's research." *Current Alzheimer research* 2:37-45. http://www.ncbi.nlm.nih.gov/pubmed/15977988.

Yildiz, Ahmet, Michio Tomishige, Arne Gennerich, and Ronald D Vale. 2008. "Intramolecular strain coordinates kinesin stepping behavior along microtubules." *Cell* 134:1030-41. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2613771&tool=pmcentrez&rendertype=abstract.

Zeitlinger, Julia et al. 2007. “RNA polymerase stalling at developmental control genes in the Drosophila melanogaster embryo.” *Nature genetics* 39:1512-6. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2824921&tool=pmcentrez&rendertype=abstract.