

**WHAT IS THE RELATIVE TIMING BETWEEN
MYOSIN'S POWERSTROKE AND PI-RELEASE?**

A Dissertation Presented

by

BRENT SCOTT

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2022

Kinesiology

© Copyright by Brent Scott 2022

All Rights Reserved

WHAT IS THE RELATIVE TIMING BETWEEN MYOSIN'S POWERSTROKE AND PI-RELEASE?

A Dissertation Presented

by

BRENT SCOTT

Approved as to style and content by:

Ned Debold, Chair

Wouter Hoogkamer, Member

Jimi Oke, Member

Marco Capitanio, Member

Richard Van Emmerik, Department Chair
Kinesiology

DEDICATION

To Megan, Marilyn, Mollie, Anna, and Amy

ABSTRACT

WHAT IS THE RELATIVE TIMING BETWEEN MYOSIN'S POWERSTROKE AND PI-RELEASE?

SEPTEMBER 2022

BRENT SCOTT

B.S., BELMONT UNIVERSITY

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Ned Debold

Myosins are a family of motor proteins responsible for various forms of cellular motility, including muscle contraction and vesicular transport. The most fundamental aspect of myosin is its ability to transduce the chemical energy from the hydrolysis of ATP into mechanical work, in the form of force and/or motion. A key unanswered question of the transduction process is the timing of the force-generating powerstroke relative to the release of phosphate (Pi) from the active site. We examined the ability of single-headed myosin Va to generate a powerstroke in a single molecule laser trap assay while maintaining Pi in its active site, by either elevating Pi in solution or by introducing a mutation in myosin's active site (S217A) to slow Pi-release from the active site. Upon binding to the actin filament, WT myosin generated a powerstroke rapidly (500 s^{-1}) and without a detectable delay, both in the absence and presence of 30 mM Pi. The elevated levels of Pi did, however, affect event lifetime, eliminating the longest 25 percent of binding events, confirming that Pi rebound to myosin's

active site and accelerated detachment. The S217A construct also generated a powerstroke similar in size and rate upon binding to actin despite the slower Pi release rate. These findings provide direct evidence that myosin Va generates a powerstroke with Pi still in its active site. Therefore, the findings are most consistent with a model in which the powerstroke occurs prior to the release of Pi from the active site.

TABLE OF CONTENTS

	Page
ABSTRACT	v
LIST OF TABLES	viii
LIST OF FIGURES.....	ix
 CHAPTER	
1. INTRODUCTION	1
1.1 Historical Significance of the Cross-Bridge to Muscle Physiology and Kinesiology	1
1.2 The Modern Cross-Bridge Cycle	3
1.3 Powerstroke or P_i -Release? a biophysicists “chicken-or-egg” causality dilemma	4
1.4 Q: How can the dilemma be resolved? A: By rigorously testing the current “unifying” theory set forth by structural biologists	5
1.5 Problem: the powerstroke is FAST. How do I see it? Solutions: Build a better mouse trap or get a slower mouse.....	7
1.6 Two steps forward, one step back	8
1.7 Specific Aims	9
1.7.1 Aim 1: Develop software to automate the analysis of laser trap data	9
1.7.2 Aim 2: Test the “slower mouse”. Determine if the S217A mutation in myosin V has a reduced displacement or rate of its powerstroke.	10
1.7.3 Aim 3: Test the “better mouse trap”. Determine if fast chicken skeletal muscle myosin II has an altered displacement or rate of its powerstroke under high levels of P_i by analyzing data from the UFFC.	11
2. LITERATURE REVIEW	13
2.1 Myosin Structure	13

2.1.1	How does P _i leave the active site?	14
2.1.2	How does the powerstroke occur?	17
2.2	The Debate	21
2.2.1	Evidence for the Powerstroke Preceding P _i -release	21
2.2.2	Evidence for P _i -release Gating the Powerstroke	23
2.3	Methodological Limitations	24
2.3.1	Structural limitations	25
2.3.2	Functional limitations & the “unifying” hypothesis	26
2.4	The S217A mutation in myosin V	27
2.5	Analysis of single molecule trap data	28
2.5.1	Manual identification	30
2.5.2	Variance threshold	31
2.5.3	Correlated thermal diffusion	34
2.5.4	Mean Variance Analysis	35
2.5.5	Page Method	38
2.5.6	Hidden Markov Model	40
2.5.7	Ensemble Averaging	44
2.6	Ultra-fast force clamp (UFFC)	46
3.	METHODS	51
3.1	Protein Isolation and Expression	51
3.1.1	Myosin V expression	51
3.1.2	Skeletal muscle myosin II isolation	51
3.1.3	Actin isolation and labeling	52
3.2	Laser trap assay	53
3.2.1	Analysis of laser trap data	54
3.3	Ultra-fast force clamp	54
4.	LASERTRAPR: A COMPUTATIONAL TOOL FOR AUTOMATING THE ANALYSIS OF LASER TRAP DATA	57
4.1	Introduction	57
4.2	Results & Discussion	60

4.2.1	Single Molecule Analysis Validation: Simple	60
4.2.2	Single Molecule Analysis Validation: Adding Complexity	63
4.2.2.1	Simulating displacement distributions	63
4.2.2.2	Simulating attachment time distributions	64
4.2.2.3	Short Events	68
4.2.3	Accuracy of determining beginning and end of events	68
4.2.4	Ensemble Averaging	71
4.2.4.1	First attempts	73
4.2.5	Testing the Ensemble Averager	76
4.2.6	Mini-Ensemble Analyzer	82
4.2.7	Single Molecule Event Identification (step-by-step)	84
4.2.7.1	Raw data	84
4.2.7.2	Processed Data	84
4.2.7.3	Running Mean & Variance	85
4.2.7.4	Hidden-Markov Model	87
4.2.7.5	Measure Events	93
4.2.7.6	Direction Correction	96
4.2.7.7	Changepoint Analysis	96
4.2.7.8	Direction Correction	102
4.2.7.9	Changepoint Analysis	102
4.3	User Manual	106
4.3.1	Getting Started	106
4.3.2	Install R & RStudio	107
4.3.3	Download <code>{lasertrapr}</code>	107
4.3.4	Launch the App	107
4.3.5	Upload Data	108
4.3.5.1	The <code>~/lasertrapr/</code> folder	108
4.3.5.2	Create Folders	109
4.3.5.3	Load Data	109
4.3.6	Clean and Process	111
4.3.6.1	Clean	111
4.3.6.2	Process	112
4.3.6.3	To Include, or not to Include.....	114
4.3.6.4	Save!.....	114
4.3.7	Analyzers	114

4.3.7.1	HM-Model/Changepoint	114
4.3.7.2	Mini-Ensemble	116
4.3.8	Summarize	116
4.3.9	Ensemble Average	117
4.3.9.1	Prep Ensembles	117
4.3.9.2	Average & Fit Ensembles	119
4.3.9.3	Plot Ensembles	120
4.4	Lasertrapr Cloud: Beta	120
5.	MYOSIN'S POWERSTROKE OCCURS PRIOR TO THE RELEASE OF PHOSPHATE FROM THE ACTIVE SITE	122
5.1	Introduction	122
5.2	Results	127
5.3	Discussion	137
5.3.1	The powerstroke precedes P_i -release from the active site	137
5.3.2	The myosin Va powerstroke occurs in two phases	143
5.3.3	What is the fate of the cross-bridge after P_i rebinds	144
5.4	Conclusions	147
5.5	Supplemental Material	148
5.6	Detecting removal of the hitch	150
5.7	What state is S217A detaching from?	154
5.7.1	Exponential fits to attachment times	156
6.	ULTRA-COOL CONCLUSION	160
6.1	The “cherry-on-top”	160
6.2	Direct observation of myosin’s powerstroke	161
6.3	Ensemble averages	163
7.	THE END	165
8.	EPILOGUE	166
Bibliography		167

LIST OF TABLES

Table	Page
4.1 App correctly identified the dispalcement size, attachment times, and time between events	63
5.1 1-exp parameter estimates	159
5.2 2-exp parameter estimates	159
5.3 3-exp parameter estimates	159

LIST OF FIGURES

Figure	Page
1.1 A typical model of the cross-bridge cycle. The powerstroke and release of P_i are often represented as occurring within the same step.....	5
2.1 Myosin Structure from Robert-Paganin 2020. The three main conserved sites are highlighted in red boxes. ATP is shown in a dashed circle.	14
2.2 The so-called ' P_i -release tunnel' detailed by X-Ray Crystallography from Llinas et al. 2015.	16
2.3 Overlayed structures of myosin V in the pre-powerstroke and rigor state from Wulf 2015.....	18
2.4 Additional lever arm rotation with ADP Release from Wulf 2015.....	20
2.5 Diagrammatic explanation of the correlated thermal diffusion analysis from Mehta et al. 1997.	35
2.6 MV histograms constructed at different window width from Guilford et al. 1997	38
2.7 Event identification via the Page Method from Knight 2001	41
2.8 Hidden Markov Model identifies single molecule events from Smith et a. 2001	43
2.9 Ensemble averaging data from Capitanio 2006.	45
2.10 The Ultra-Fast Force Clamp from Capitanio 2012.....	50

4.1	A simulated laser trap event. The event has 200ms of baseline preceding a 5nm displacement that lasts for 100ms. Another 200ms of baseline data is appended after the displacement. This exact sequence was replicated and concatenated together 200 times which yields a data trace with 200 simulated binding events that are spaced 400ms a part. The exact simulated measurements are: 5nm displacement, 100ms attachment times, and 400ms time between events.	61
4.2	Screenshot of the output from the single molecule analysis produced by lasertrapr . Black lines is the original trap trace, the green line signifies the results of the hidden markov model performed on the running window transformation, and the yellow highlights is the final results after changepoint is applied.	62
4.3	Four simulations were performed with average dispalcements of 0, 5, 10, and 15nm. a) Simulated data traces. Top trace has 0nm average displacement. Bottom has a 15nm average displacement. b) Linear regression of the actual displacement (x-axis) versus the analysis estimated displacement size (y-axis) demonstrating a near perfect correlation. c) Full gaussian distributions of the analyzer's estimated displacement for each of the 4 simulations.	65
4.4	Simulated truncated (a & b) exponential distributions that are representative to the ones used to generate the simualted data	67
4.5	Four simulations were performed with average attachment lifetimes of 50....	69
4.6	Simulated data trace with very short events. Measured attachment times for the 3 ID'd events in the trace are 18, 15, and 32ms. Plot is interactive online. Grid lines represent 1s and 2nm intervals....	70
4.7	Analysis of the misalignment in the true start/end of an event and the analysis estimated start/end of an event. a) About 75% of the start of events are picked within several datapoints. b) The ends of events are mostly picked with a couple data point error. c) A table showing the average or median number of datapoints that the analysis missed the start or end of an event by.	72
4.8	Ensemble averages created for Gunther 2020. Blue is WT mysoin. Red is the S217A mutation. All data collected at pH 7.0 and 0mM-P _i . Raw traces are inset and ensembles are fit a with single exponential.	75

4.9	Ensmebles averages of the simulated datasets for validation. The tables show values from the fits that closely correspond to the true values.	81
4.10	An early output plot from the mini-ensemble analyzer. The plots are now scrollable/interactive, but this serves as a fun reminder of where the app began in its infancy years prior to its creation.	83
4.11	Image from Woodward et al. (2020) showcasing the mini-ensemble analyzer that is now a core feature within the app.	84
4.12	Raw trap data.....	85
4.13	Example of a step calibration. Stage was moved 200nm	86
4.14	Example of processed trap data	86
4.15	Mean/Variance Plot	88
4.16	Mean/Variance Plot colored by State	92
4.17	1ms running variance transformation	97
4.18	Changepoint identified start of event in running variance	98
4.19	Changepoint identified start of event in the original data set	99
4.20	Results of analysis overlayed on data trace	100
4.21	1ms running variance transformation	103
4.22	Changepoint identified start of event in running variance	104
4.23	Changepoint identified start of event in the original data set	105
4.24	Results of analysis overlayed on data trace	106
5.1	Single molecule laser trap assay data and analysis.	128
5.2	Ensemble averaged binding events from Scott Marang 2021.	131
5.3	Binding event durations and bulk assays measurements.	135
5.4	Proposed model of cross-bridge cycle. A simplified model postulating the effects of P _i and the S217A myosin's cross-cycle.	147

5.5	Conceptual models of the simulated data.....	148
5.6	Step size comparison. No significant differences found in the average displacement between two simulated conditions with and without the hitch. Plot is interactive online.....	153
5.7	Exponential fits to the cumulative distributions of attachment times. The observed data is in the thick grey line. 1-, 2-, and 3-component exponential fits are shown. Plot is interactive online.....	158
6.1	Raw data trace from the UFFC showing the initial binding of myosin to actin (blue), direct observation of the powerstroke (green), and ATP induced dissociation (red). Data was collected at 1mM-ATP and 30mM-Pi. Time scale is presumably in seconds so the entirety of the event is ~2ms.....	162
6.2	There are 5 single molecule displacements in this 2 second simulation that are between 1-10ms in duration. Can you spot them all?.....	162
6.3	A possible conceptual representation of a single UFFC event under a P_i -release first model in high P_i concentration conditions. This effect was not observed.	163
6.4	Ensemble average of UFFC data at 30mM-Pi. Myosin's powerstroke occurred rapidly within several hundred micro-seconds even under high P_i conditions and substantial resistive load.	164

CHAPTER 1

INTRODUCTION

1.1 Historical Significance of the Cross-Bridge to Muscle Physiology and Kinesiology

A fundamental aspect of Kinesiology is centered upon understanding the mechanisms of human motion. Ultimately, every human movement is a direct result of a molecular motor (like myosin) transducing the chemical energy of adenosine triphosphate (ATP) into a mechanical force. Understanding how human skeletal muscle can accomplish the coupling between biochemistry and mechanics has transcended any single field of science over the past century and has required a multi-disciplinary approach to reach the current understanding we currently have on the topic.

In the early 1900s, pioneers in the field of muscle physiology provided the foundational work in understanding how muscles work. During this time the field of biochemistry was unclear of what the exact mechanisms were that provided a muscle cell with the necessary energy for muscle to contract, which was often contributed to oxygen (or at least some form of an oxidative pathway) (Bassett 2002) or to “lactic acid” (Herzog et al. 2015). In 1920, A.V. Hill observed the liberation of heat during isometric tension of an isolated muscle (Hill and Hartree 1920) showing that muscular force can be produced without oxygen. His findings ultimately led him to a Nobel Prize in 1922 (“The Nobel Prize in Physiology or Medicine 1922,” n.d.). This finding was detailed further when Wallace Fenn demonstrated that when allowed to shorten, a contracting muscle liberates more heat than when held isometrically (now well known as the Fenn effect) (Fenn 1924). Subsequently, Bailey (1937) characterized

the abundance of myosin in skeletal muscle and Engelhardt and Ljubimowa (1939) demonstrated that myosin was indeed responsible for the muscle's enzymatic activity and consequently the heat production that was observed by Hill and Fenn during their isolated muscle studies years prior. However, despite these early advances there was still no explanation for how a muscle (let alone a myosin molecule) could use ATP to produce force.

The next major breakthrough propelling the field closer to an answer for the mechanism of molecular force production came in 1954 when a pair of Nature articles from Huxley & Hanson and Huxley & Niedergerke independently provided evidence for muscle contraction via the “sliding filament theory” (H. Huxley and Hanson 1954; A. F. Huxley and Niedergerke 1954). Here, the two research groups describe the ability of myosin in the form of thick-filaments to associate with actin filaments causing sarcomere shortening (i.e. contraction) based on structural observations of contracting myofibrils and single muscle fibers. Even so, the sliding filament theory in of itself is not a mechanism of contraction. In 1957, Andrew Huxley provided the first mechanistic hypothesis about how the relative sliding of filaments could occur based upon modeling myosin as a biased Brownian ratchet. This proposal provided the initial groundwork for the modern-day “cross-bridge theory” of muscle contraction (HUXLEY 1957).

After the emergence of these structural perspectives of the actomyosin cross-bridge system biochemists began attempting to align their observations from solution kinetic studies to further explain the structure function relationship of the cross-bridge providing early kinetic schemes of the rates of the various mechanochemical steps/states of the cross-bridge model (Lynn and Taylor 1971). Since the completion of these foundational studies research into the field of muscle physiology has exploded as technology has advanced temporal and spatial limitations of instrumentation providing unprecedented details into muscle myosin structure and function which has provided

a relatively deep understanding (versus the humble origins) of how the cross-bridges cycle from both a biochemical and structural perspective.

1.2 The Modern Cross-Bridge Cycle

The modern cross-bridge cycle is a mechanochemical system which describes the coordination between myosin's enzymatic biochemistry and structural conformations. Different structural conformations are related to certain biochemical states and influence myosin's affinity for the hydrolysis products and actin. As such, the cross-bridge cycle can be modeled as a summary of the current biochemical and structural conformations with a basic model being:



where A is actin, M_{post} is myosin in a post-powerstroke conformation, M_{pre} is myosin in a pre-powerstroke conformation, T is ATP, D is ADP, and P is P_i .

1) Starting in rigor, myosin occupies a post-powerstroke position strongly bound to actin (Geeves and Holmes 1999; Sweeney and Houdusse 2010). 2) The binding of ATP to myosin causes a structural rearrangement in the active site which ultimately opens the upper and lower 50-kDa (Conibear et al. 2003; Holmes et al. 2003) causing both the detachment from actin and resetting of the lever arm into a pre-powerstroke position (Nesmelov et al. 2011; Trivedi et al. 2015) . 3) ATP hydrolysis occurs constraining the myosin in the pre-powerstroke position. 4) When myosin attaches to actin there is a short lived strongly bound/load bearing state where myosin is pre-powerstroke with both hydrolysis products in the actin site (Llinas et al. 2015; Woody et al. 2019) . 5) The final step in the cycle is where myosin undergoes a powerstroke (i.e. lever arm rotation) and P_i -release which leaves the molecule in a

strongly bound post-powerstroke state with solely ADP occupying the active site. Once ADP is released, myosin is left in rigor and the cycle restarts.

1.3 Powerstroke or P_i-Release? a biophysicists “chicken-or-egg” causality dilemma

While there are many details in the modern day cross-bridge cycle that are seemingly clear and well-established, there is a serious point of contention that remains in the literature to this day. The problem resides in the transition between the 3rd and 4th step in the model detailed in section 1.2. In **one** step, **two** key events are taking place - one mechanical and one biochemical. The mechanical powerstroke and biochemical release of P_i are collapsed into this single step without an explicit declaration of the timing or sequence relative to one other. **Both these events are triggered by actin binding, but does the powerstroke precede P_i-release or does P_i-release “gate” the powerstroke?**. This *is the dilemma*. Interestingly, the Powerstroke First and P_i-Release First models are *both* well supported in the literature. The divergence from a common mechanochemical scheme has divided the field for several decades and has ultimately limited the progress of fully understanding how myosin is able to transduce chemical energy into mechanical work as the powerstroke and P_i-release steps are key events in the transduction process. Ultimately, not understanding the precise order of the biochemical and mechanical events during the cross-bridge cycle prevents progress into the mechanisms underlying the molecular nature of energy conservation (1st law of thermodynamics) and limits our ability to develop pharmacological interventions to restore function to diseased, mutated, or compromised motors (HCM, DCM, fatigue, etc.). Moreover, in the emerging field of nanotechnology where researchers have been increasingly more interested in controlling biological motors understanding the precise ordering of steps in the transduction is paramount to the success of the field. For example, if you wanted to control myosin

by preventing the powerstroke from occurring (and assuming a P_i -gating model of the cross-bridge) force production could be halted by trapping P_i in the active site. However, this approach would not work in a Powerstroke First model.

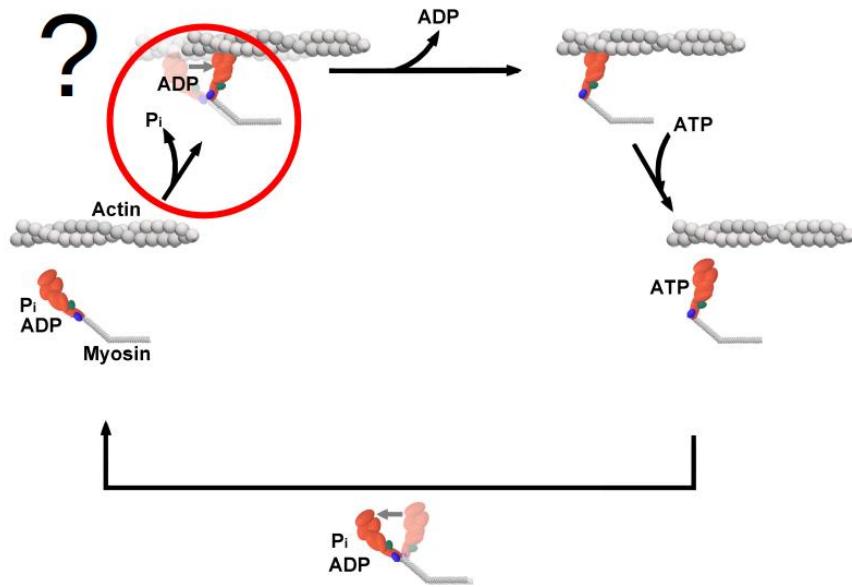


Figure 1.1. A typical model of the cross-bridge cycle. The powerstroke and release of P_i are often represented as occurring within the same step.

1.4 Q: How can the dilemma be resolved? A: By rigorously testing the current “unifying” theory set forth by structural biologists

In 2015, Llinas et al. (2015) performed x-ray crystallography on phosphate soaked myosin crystals and reported a new and never-before-seen myosin conformation that was posited to provide a unifying theory between the Powerstroke First and P_i -Release First models. This conformational state was aptly named the P_i -Release State (P_iR State). Accompanying the P_iR state, they provided a hypothesis regarding the structural sequence of P_i being released from myosin: 1) Myosin is in a Pre-powerstroke

state (PPS) with P_i in active site, 2) the release of P_i from the active site into the so-called “phosphate release tunnel” causes slight re-arrangement into the P_iR2 state (this is Powerstroke gating in their model), and 3) the final crystal structures detail P_i in the P_iR1 state as the P_i is released into solution from the P_i -release tunnel. The subtly is that there are **two** different P_i -releases. A phosphate can be released from the active site into the P_i -release tunnel as well as be released from the P_i tunnel in myosin out into solution. Moreover, they state that the P_i -release from the active site into the P_i -release tunnel (PPS to P_iR2 transition) is what gates the powerstroke and not the release from the tunnel into solution (see literature review for more details). Now, this is an elegant hypothesis that would explain away some of the most convincing functional data that supports the Powerstroke First model. Mainly, in coupled assays where the force of a muscle fibers is measured simultaneously with the rate of P_i -release (with a phosphate binding protein), or in the case with solution kinetic studies and FRET, the rate of the powerstroke is always measured to be far greater than any measured P_i -release rate. However, with the P_iR state the structural biologists argue that a phosphate binding protein would be binding P_i in solution with the equivalent myosin structural state being P_iR1 or later. Aka, the structural biologists are saying that an assay where a phosphate binding protein is used to measure the rate of P_i -release is actually measuring the wrong rate which makes the previous reports where the powerstroke was found to occur at faster rates an unfair comparison and that indeed the P_iR2 to P_iR1 is rapid and able to gate the powerstroke. However, this current “unifying” theory set forth in recent years remains to be rigorously tested and provides a testable hypothesis - *If the transition of P_i from the active site into the P_iR2 position gates the powerstroke, then can the powerstroke be prevented or slowed by occupying the active site with P_i ?*

1.5 Problem: the powerstroke is FAST. How do I see it?

Solutions: Build a better mouse trap or get a slower mouse.

While the question is simple, “can the powerstroke be prevented or slowed by occupying the active site with P_i ?”, devising a way to test this is not so simple for one main reason - the powerstroke occurs quite rapidly upon binding to actin (upwards of 5000/s) (Marco Capitanio et al. 2012; Woody et al. 2019). Being able to assess whether or not experimental conditions can affect the powerstroke is quite challenging. Ideally, these experiments will be conducted in a 3-bead laser trap assay which allows for the observation of single actomyosin binding events and through further analysis the displacement of every individual event can be measured and average displacements can be calculated and compared amongst the conditions. Furthermore, after the identification of the individual events, the trapping records can be ensemble averaged in order to measure the averaged rate of the transition from unbound to bound states. Experimentally, there are two ways that can be used to occupy myosin’s active with P_i . 1) By increasing the concentration of free P_i in solution the probability of P_i rebinding into the active site is increased, or 2) by using a mutation that prevents the release of P_i from the active site into the P_i -release tunnel (the S217A mutation in myosin V does just this) (Forgacs et al. 2009; Llinas et al. 2015). Really, the S217A mutation is just a “slower mouse” and affords the ability to use standard trapping techniques in order to test the “unifying” theory. The S217A mutation in myosin V has been shown to have an impaired rate of P_i -release of 30/s Llinas et al. (2015) which is far slower than the WT control myosin V (200/s). An event occurring at 30/s would on average take 33 milliseconds to be completed. If the release of P_i from the active does indeed gate the powerstroke there should be long delays between S217A myosin binding actin and the powerstroke (~33 milliseconds) which would readily be observable under a standard optical trapping setup that has

millisecond time resolution. However, most WT myosins still have a relatively fast P_i-release rate when compared to the resolution of a standard optical trap which might still make the determination of the order of these events difficult. To overcome this, the Ultra-Fast Force Clamp (UFFC) has been used previously which provides sub-millisecond time resolution and allows for the observation of weak binding, an initial force bearing state, and the powerstroke itself in individual actomyosin binding events Woody et al. (2019). More simply, the UFFC is the “better mouse trap”.

1.6 Two steps forward, one step back

As presented there is seemingly a couple plausible experiments that can be performed to further assess the “unifying” theory of the powerstroke-first and Pi-release first model. However, there are currently not **any** free open source software (FOSS) projects whose goal is specifically for the automation of single molecule laser trapping data. This (perhaps) is one of the biggest obstacles that is limiting the field of single molecule biophysics. Without a community of software developers there are no standards for the analysis of our data and every new scientist (like myself) has to re-invent the wheel (or at least some version of it) from what they can piece together from the (sometimes *very* minimal) information provided by the methods section of an article. Unfortunately, I have no quantitative evidence supporting the limitations that a lack of analysis software is imposing on the field, but the evidence I do have is personal and anecdotal. This dissertation. As a graduate student I have identified and defended a research topic through my comprehensive exams and have even outlined in previous sections (see 1.4, 1.5) some of the scientific/experimental approaches that could serve as foundation to help progress the knowledge of the field on this given topic. However, even if I was handed all of the data needed to complete the project, I would not be able to answer the questions. Why? Because I do not have the right analysis programs/tools. Generally there are two common methods for an-

alyzing single molecule laser trap data, 1) single molecule event identification (event picking) and 2) Mean Variance (MV) analysis (see Literature Review for more details on single molecule analysis techniques). Our lab primarily uses MV for analyzing single molecule data (Woodward et al. 2020; Unger and Debolt 2019) and has had success with this method. While there are pros to using MV (see Literature Review) it may not be the best candidate to answer my specific questions. Mainly, MV does not identify individual events and as a result additional post-analyses cannot be run on the data, like ensemble averaging. With MV, a change in the magnitude of the displacement of the powerstroke could be estimated, but the rate of this transition could not be determined. This is a key question that needs to be answered for this project to be successful and so an event picking analysis program is needed. But, there are no available programs that implement this, that could be used easily, or that is FOSS. Sure, plenty of research groups have their *own* programs, but the source code is not available and details are limited (some authors do not even mention software/programming languages used) about how specific analyses work which makes replicating an analysis tricky and quite difficult and imposes a serious limitation to the **reproducibility** of work in the field.

1.7 Specific Aims

1.7.1 Aim 1: Develop software to automate the analysis of laser trap data

The current analysis that was in use by our lab (MV) was not sufficient to answer the questions presented here and answering these questions will ultimately help progress our understanding of how myosin converts chemical energy into mechanical work. An analysis program will be developed that can identify single molecule actomyosin interactions from raw trapping data with the ability to perform ensemble averaging in order to estimate the rate of the powerstroke (transition from unbound to bound). Moreover, the program will have a user-friendly graphical user interface

(GUI) and be free and open source. The analysis will be written in the R programming language (which in of itself is FOSS) and the user interface built with Shiny (an R package that provides a web application framework). With both the analysis program and R language being FOSS, others may contribute/customize the analysis to their needs and improve the operating standards of single molecule trap analysis by offering community developed analysis solution. Additionally, this program will offer a practical, yet advanced, set of tools for other graduate students and research groups to easily implement in their own respective work. “Under the hood” the program will implement the current “best practices” of single molecule event detection by being able to access a robust network of R add-on packages written and developed by experts in their respective field (these R packages of course being well documented and well cited). Lastly, the program should not just be for analyzing single laser trap data records, but should be in fact a complete data analysis and data management tool that will offer easy-to-use tools to perform calibrations, data cleaning and management, analysis, as well as being able to perform project summary statistics. In essence, this program will be a tool that will completely automate the analysis of laser trap data from raw data to final figures.

1.7.2 Aim 2: Test the “slower mouse”. Determine if the S217A mutation in myosin V has a reduced displacement or rate of its powerstroke.

S217A is an ideal candidate to test the sequence of events surrounding the timing of the powerstroke and P_i-release in the 3-bead laser trap assay. The main advantage being that S217A effectively traps the -phosphate in the active site which has been shown to have a drastic effect on slowing the P_i-release rate as previously measured in solution kinetics studies (~30/sec). Our hypothesis is that if P_i-release gates the powerstroke, we should observe a prolonged (~33 milliseconds) delay (time period characterized by a decrease in variance of the trapping signal with no displacement)

between the initial binding of myosin to actin and occurrence of the powerstroke. The resulting effects of the S217A mutation would be a reduced displacement (leftward shift in the displacement distribution compared to WT) and a slowed transition rate in the ensemble averaged data. However, if the powerstroke precedes P_i -release then the S217A displacement distributions, and ensemble averages should look identical to the WT. Additionally, performing this same experiment again with high (30mM) P_i in solution will provide an alternative methods for attempting to occupy the active site with P_i . As such, our hypothesis is that if P_i gates the powerstroke there should be a prolonged delay between the initial binding and powerstroke (but much longer in time for S217A vs WT due to slowed P_i -release rate). If the powerstroke precedes P_i -release then all 4 conditions WT 0mM- P_i , WT 30mM- P_i , S217A 0mM- P_i , and S217A 30-mM- P_i will have identical displacement distributions and ensemble averages.

1.7.3 Aim 3: Test the “better mouse trap”. Determine if fast chicken skeletal muscle myosin II has an altered displacement or rate of its powerstroke under high levels of P_i by analyzing data from the UFFC.

While the S217A provides an ideal test since the mutation has drastic affects on slowing the P_i -release rate, there are still limitations imposed by the setup of the standard optical trapping technique that prevent direct observation of myosin binding actin and the subsequent powerstroke. UFFC provides an increased time resolution and allows for the direct observation of the rate of a single powerstroke by a single molecule of myosin interacting with a single actin filament. This is in contrast to inferring an averaged “powerstroke” rate from the ensemble averaged data collected under a standard trapping protocol. Furthermore, the UFFC has the necessary time resolution in order to test a faster myosin isoform (chicken fast skeletal muscle myosin II) under no (0mM) and high (30mM) P_i which will provide additional data on a

different myosin isoform helping to provide a more robust and inclusive answer to the proposed question. Our hypothesis is if P_i -release gates the powerstroke there should be a prolonged initial weak binding state between the binding of myosin to actin and the powerstroke under 30mM P_i compared to 0mM P_i , but the rate of the actual powerstroke should be unchanged.

CHAPTER 2

LITERATURE REVIEW

This review is aimed to be a comprehensive examination of the literature with a scope highly focused on the aspects most fundamental to the main question addressed by this dissertation (what is the relative timing of the powerstroke and P_i -release from myosin?). The review will begin by briefly covering myosin structure and move specifically into how P_i is released from the active site and how lever arm rotation occurs before discussing the evidence in-support of both the Powerstroke First and P_i -Release First models. Additionally, methodological limitations will be addressed here as this may/may not confound the conclusions drawn from a report. Lastly, an evaluation of common analysis techniques of single myosin molecule laser trap will be given including an emphasis on the lack-of availability of user friendly software.

2.1 Myosin Structure

Myosin is a motor protein that accomplishes a wide variety of cellular tasks that includes both muscle contraction (myosin II) and intracellular cargo transport (myosin V). In fact, the myosin (super) family is quite large and has >100 different myosin isoforms (Hodge, Jamie, and Cope 2000) that have traditionally been classified into >30 different structural classes. However, more recently there has been an effort to more quantitatively classify myosins into four groups based on mechanic and kinetic parameters. This grouping schema classifies a myosin as either a 1) fast mover, 2) slow/efficient force holder, 3) strain sensor, and 4) gate (Bloemink and Geeves 2011). Despite a large diversity of job that each myosin performs, they all are mechanochem-

ical enzymes with conserved structures which allows them to transduce the chemical energy of ATP into a mechanical force imposed onto an actin filament. The conserved structures in the myosin motor domain contains 3 distinct sites which interact together allowing the enzymatic (biochemical) and mechanical function of the whole protein to occur. The sites are the 1) active site, 2) the actin binding surface, and 3) the lever arm (Robert-Paganin et al. 2020). Structural changes that occur at both the actin binding surface and in the active site ultimately are communicated throughout the entire protein which effects the position of the lever arm.

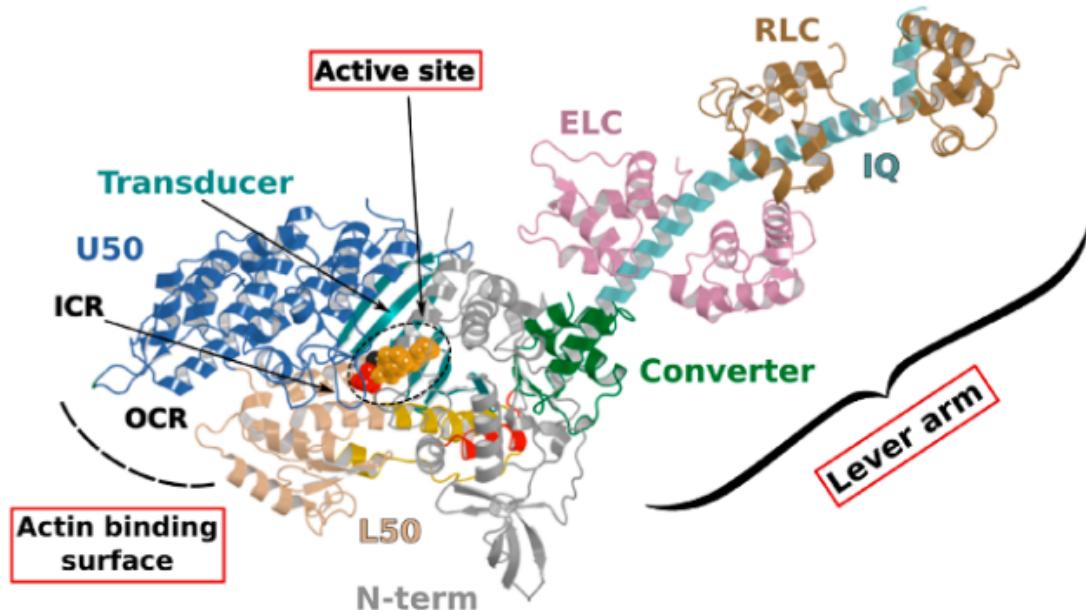


Figure 2.1. Myosin Structure from Robert-Paganin 2020. The three main conserved sites are highlighted in red boxes. ATP is shown in a dashed circle.

2.1.1 How does P_i leave the active site?

When binding to myosin, an ATP molecule enters the active site “-phosphate first”. The implications of this is that when ATP is hydrolyzed to ADP + P_i , the ADP molecule blocks the P_i exit pathway the way that ATP entered. Counterintuitively, P_i -

release occurs first prior to the release of ADP in myosin's cross-bridge cycle (Geeves, Goody, and Gutfreund 1984). But, how does P_i leave the active site if ADP is blocking the exit? There are three proposed pathways that lead into myosin's active site and they are nicknamed the "front door", "back door", and "side door". The position of ATP in the active site was first observed when myosin's crystal structure was first resolved (Rayment et al. 1993). At this time solution kinetic studies had established the timing of some of the biochemical steps and was clear that the P_i -release step occurred before the release of ADP. With the observation of ATP in the active site " -phosphate first" it became clear that P_i could not leave the same way it originally entered (via the "front door") due to steric blocking by ADP. With this data the hypothesis was made that myosin could be a "back door" enzyme in which P_i could escape through the 50-kDa cleft (Gilson et al. 1994; Rayment et al. 1993; Yount, Lawson, and Rayment 1995). Additionally, over time other hypotheses arose most prominently with the idea of P_i leaving through a "side door" which forms from an opening in between SWII and the P-loop from the active site. To date, the most probable P_i -release pathway is via the "back door" and is thoroughly detailed by Llinas et al. (2015). Using x-ray crystallography, myosin-VI, varying levels of $[P_i]$ concentrations, and numerous active site point mutations they characterized a so-called " P_i -release tunnel" in between the cleft of the two 50-kDa domains that lead from the active site into solution. In short, the mechanism of P_i -release involves actin binding triggering conformational changes in the active site. SWII opens the entrance to the P_i -release tunnel in which the P_i can dissociate from the active site into solution. Additionally, they were able to couple their structural observations with kinetic solutions studies to show the higher probability of the "back door" mechanism versus the alternate "side door" hypothesis.

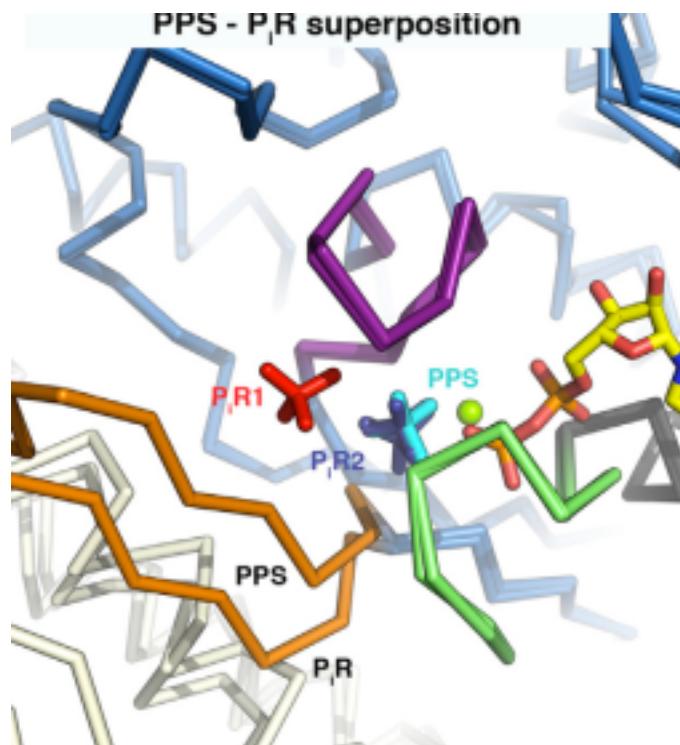


Figure 2.2. The so-called 'P_i-release tunnel' detailed by X-Ray Crystallography from Llinas et al. 2015.

2.1.2 How does the powerstroke occur?

Unfortunately, there are no structures of the transition states that occur throughout the cross-bridge cycle and thus there is limited knowledge of the structural rearrangements that occur during the main part of the powerstroke (the force generating step), however what is known is that there is a large lever arm rotation between the initial force bearing P_iR state and the strongly bound ADP state that is coupled with cleft closure (Sweeney, Houdusse, and Robert-Paganin 2020). Structural data has however provided information about the size of the lever arm, ~9nm in chicken skeletal S1 (Rayment et al. 1993), which is in close proximity with single molecule displacement records measured in the laser trap, 5-10nm, for skeletal myosins containing two IQ domains Tanaka et al. (1998). In addition, the displacement of the powerstroke scales with the length of the lever arm which is dependent on the number of IQ domains (Matthew J. Tyska and Warshaw 2002). In comparison, myosin V S1 that contains 6 IQ domains was measured to have an average displacement ~18 nm (Veigel et al. 2002). Interestingly, the displacement of myosin V S1 is about three times larger than that of myosin II S1 as myosin V S1 also contains three times the number of IQ domains.

Along with the length measurements, estimations of the degrees of rotation of the lever arm have been made. Several groups have estimated a large 65-75 degree rotation of the lever arm which occurs as result of the powerstroke (Dominguez et al. 1998; Houdusse, Szent-Györgyi, and Cohen 2000; Walker et al. 2000). Admittedly, though the estimations of the displacement caused by the powerstroke are rather close between measurements of the lever arm from structural biologists and displacement records provided by single molecule trappers, there has been speculation on why the displacement measured in the laser trap is often a few nanometers less than predicted by a 9nm lever arm rotating through 70 degrees of motion (this should be >9nm based of structural predictions though many trappers report 5-6nm). Matthew J. Tyska and

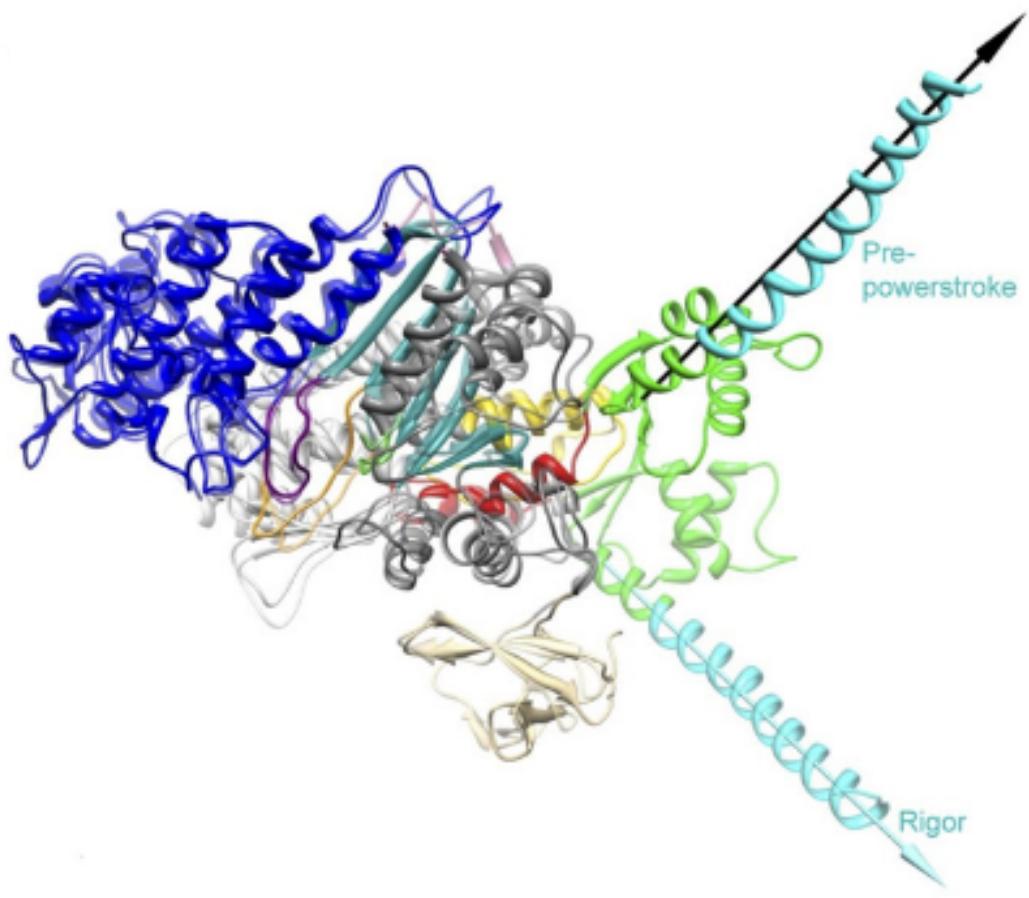


Figure 2.3. Overlayed structures of myosin V in the pre-powerstroke and rigor state from Wulf 2015

Warshaw (2002) attribute the underestimation from trapping studies to the fact that myosin is working against a load in the laser trap. Alternatively, there could also be series compliance that effects the functionally measured displacement in the trap. Furthermore, one of the most notable and important features of the myosin motor domain is the converter. The converter is able to amplify small conformational changes that occur at the actin binding interface and the active site into large movements and re-positioning of the lever arm. The converter is highly flexible and the majority of the series compliance in the myosin has been attributed to this feature (Köhler et al. 2002). Though the lever arm has been suggested to be compliant by some (Howard and Spudich 1996), more recent evidence suggest the lever arm is in fact rigid (Warshaw et al. 2000) since the observation that showed myosins displacement is directly proportional to lever arm length (Matthew J. Tyska and Warshaw 2002). Interestingly, the powerstroke actually occurs in two transitions, the fast powerstroke and the second powerstroke (sometimes referred to as the hitch). The additional lever arm rotation is accompanied by complete closure of the actin binding cleft that is coupled with the release of ADP from the active site (Mentes et al. 2018; Sweeney, Houdusse, and Robert-Paganin 2020). While, the second powerstroke does not contribute much to the development of force, it is crucial to the *maintenance* of force. Changing the rate at which ADP is released from the active site can drastically alter myosin's duty ratio and is highly load dependent. In fact, kinetic differences, including differing ADP release rates, is what make different myosins ideal at performing specific tasks. Through their kinetics myosins are essentially "tuned" to their biological function. While skeletal muscle myosin II has a low duty ratio (~5%) which makes it ideal for the development of high forces and velocities, myosin V has a much higher duty ratio which allows for one head to always be strongly bound to an actin filament track ensuring the molecule does not simply diffuse off of its actin track.

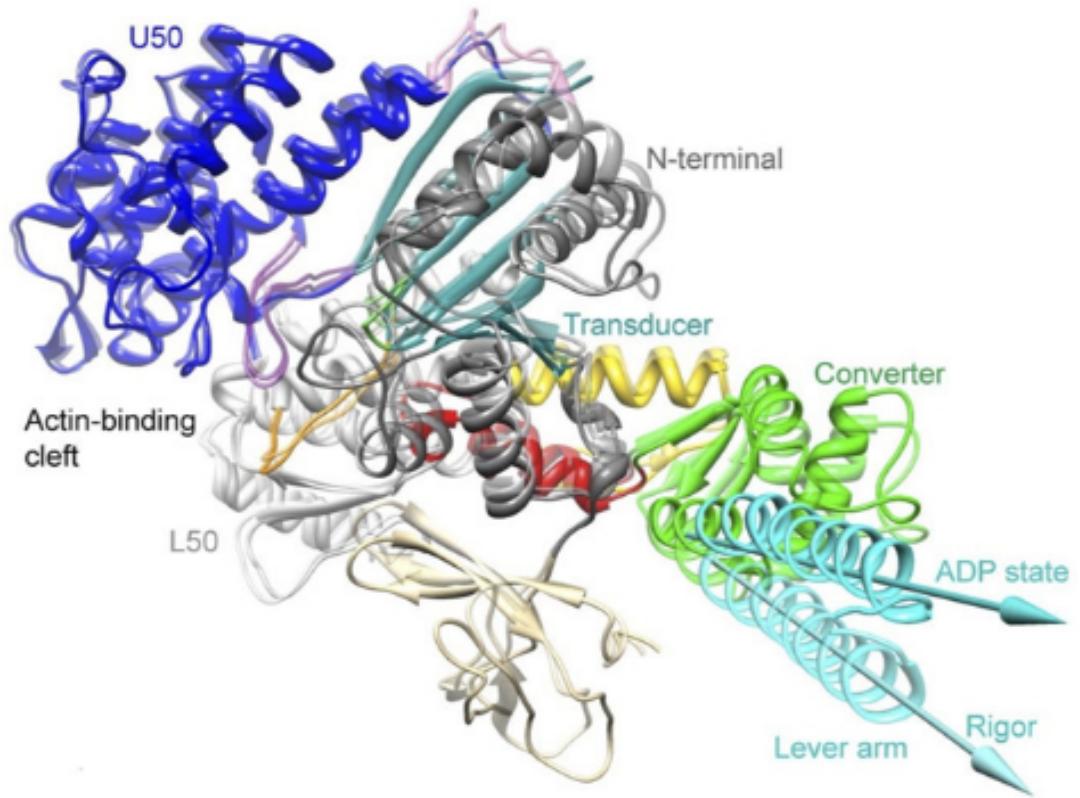


Figure 2.4. Additional lever arm rotation with ADP Release from Wulf 2015

Lastly, the key activator of the myosin powerstroke that is universally agreed upon is actin binding. Really, to try and explain what happens after actin binding would be jumping too far into the major debate happening in the field and is exactly the motivation behind this dissertation. Up until now, what is fairly certainly known (and what is actually generally agreed upon) is that the powerstroke is 1) actin-activated and 2) occurs between an initial force bearing state and the strongly bound ADP state. The details and order of events that occur has been debated for years and the answer to the question “how does the powerstroke occur?” is highly dependent on who is asked. Two distinct and opposing answers are provided by structural biology and the results of functional experiments.

2.2 The Debate

The relative timing of the powerstroke and P_i -release is the big question in the myosin world. This question has been tested and re-tested and evidence continues to build for *each* side. The perspective from structural biologists holds the view that **P_i -release gates the powerstroke** and *they have the data to back it up*. However, functional assays performed by muscle physiologists, biochemists, and biophysicists support the idea that the **powerstroke occurs prior to P_i -release** and that actin-binding alone is enough to trigger the powerstroke, and *they have the data to back it up*.

2.2.1 Evidence for the Powerstroke Preceding P_i -release

In a model where myosin’s powerstroke occurs before P_i -release there must be a couple of key functional observations that can be observed experimentally in order to support this theory. 1) The development of force or the rate of the powerstroke should be relatively faster than the measured P_i -release rate. 2) The powerstroke should be reversible and/or force production should be vulnerable to P_i rebinding as this should

either reset the myosin to an M_{pre}-D-P state and/or induce premature detachment. Early studies with muscle fibers show isometric tension is inversely proportional to [P_i] - the higher the [P_i] in solution, the lower the force (Brandt et al. 1982). Additionally, the release of caged P_i in isometrically contracting muscle fibers displays a delayed but clear depression in tension (Dantzig et al. 1992). Furthermore, a linked assay that assesses the development of muscle fiber force and the simultaneous appearance of P_i in solution proves that fiber force develops faster than the appearance of P_i in solution (He et al. 1997). Later studies show that the rate of force production following a shortening-restretch protocol is rapid and similar to the original time course of the force development from isometric tension (Sleep, Irving, and Burton 2005). These cellular level studies are clear that a muscle can produce force faster than the release of P_i and that the force developed by a muscle can be affected by P_i rebinding, all in support of a model where the powerstroke occurs before P_i-release. More recently, in vitro investigations using time resolved FRET with fast skeletal myosin II provided direct evidence that the powerstroke proceeds P_i-release by obtaining fluorescence measurements of both the rate of the powerstroke and P_i-release for direct comparison. The rate of P_i-release was measured at 30-40 s⁻¹ and the adoption of a post-powerstroke configuration (i.e. M* Mole Fraction) occurred at 350 s⁻¹ (Muretta et al. 2015). Thus, when directly measured the rate of transition from pre-post powerstroke is far greater than the rate of the appearance of P_i in solution. This finding was corroborated with traditional FRET studies of myosin-V comparing lever arm rotation rate to another independently acquired P_i-release rate (Trivedi et al. 2015). Lastly, ultra-fast optical trapping can directly measure the rate of the powerstroke from a single molecule of myosin (Marco Capitanio et al. 2012). Woody et al. (2019) measured the rate of the powerstroke under varying [P_i] concentrations and saw rapid displacements occurring within ~200us of actin binding and that the dwell time prior to the displacement was unaffected by [P_i]. Even so, by

using an ensemble averaging technique they show direct observation of powerstroke reversals (under high loads and high [P_i]) which provides single molecule evidence of the powerstroke proceeding P_i-release.

2.2.2 Evidence for P_i-release Gating the Powerstroke

If P_i-release gates the powerstroke, distinct structural conformations of myosin should exist in order to support this model. 1) When P_i is in the active site myosin should be in a pre-powerstroke position and 2) the adoption of a post-powerstroke configuration should only occur if P_i is absent from the active site. Interestingly, the hypothesis that the biochemical release of P_i from myosin's active site proceeding the powerstroke originally came as a result of first x-ray crystallization of myosin (Rayment et al. 1993) and the subsequent attempts to dock an ATP molecule into the crystal structure since the original myosin structure was crystallized without a ligand. The realization occurred after these efforts that myosin may perhaps be a “back door” enzyme (Yount, Lawson, and Rayment 1995). By extending the “back door” analogy, Yount, Lawson, and Rayment (1995) described P_i as behaving as a “door stop” as it was hypothesized that P_i itself actually prevented the powerstroke from occurring since the P_i appeared to be in a position that would prevent cleft closure (and the subsequent triggering of structural changes necessary for lever arm rotation). Indeed, for over a decade from the original structural characterization of myosin there persisted discrepancies amongst multiple hypotheses about how actin binding could possibly cause the conformational changes necessary in the active site to open the back door (Sweeney and Houdusse 2010). The two possibilities being that either SWI can create the opening for the back door whilst remaining coordinated with the ADP or SWII must somehow open the back door without permitting lever arm rotation. Llinas et al. (2015) provided new myosin x-ray crystallography structures in which they show the ability of SWII to move ~4A from its position in the pre-powerstroke

position, opening the back door escape route without substantial movement of the lever arm. This new state was aptly named the P_i -release state, P_iR . By using a P_i -soaking protocol followed by rapid freezing and subsequent crystallization, Llinas et al. (2015) demonstrated that P_i can translocate back through the P_i -release tunnel as they observed two distinct structural states in response to the rapid freezing after P_i -soaking. P_i was either at the exit of the P_i -release tunnel (P_iR1) or translocated back through the tunnel near the ADP (P_iR2). Only in response to a delayed freezing after P_i -soaking did they observe P_i completely rebinding the active site, consequently reforming the pre-powerstroke state. Thus, they provide strong evidence for P_i -release occurring before the powerstroke - when P_i is in the active site, myosin is in the pre-powerstroke state. Furthermore, when P_i -soaking post-powerstroke myosin in an ADP state, the crystal structures show that the P_i was not able to translocate back through the P_i -release tunnel to the active site. This demonstrated that myosin only adopts a post-powerstroke conformation when P_i is absent from the active site and also that P_i -rebinding can only occur prior to the powerstroke (Llinas et al. 2015). These observations are all consistent with a model in which P_i -release gates the powerstroke.

2.3 Methodological Limitations

As the brief review provided by the previous two sections highlights, there is overwhelmingly strong evidence supporting both models that describe the relative timing of the P_i -release and the powerstroke. Because of this, careful consideration about the methodical and analytical limitations should be understood if there will be any hopes of somehow reconciling the differences observed in the data. To start, the most obvious difference that is evident amongst the data is that the majority of the studies that support the powerstroke first model are all functional assays, whereas the evidence supporting the P_i gating model are largely structural. In essence, structural data provides “snapshots” analogous to the creation of a stop motion film,

whereas the functional studies provide the soundtrack to the movie. We have the two essential pieces to creation of a blockbuster film, but not the information to correctly align the audio to the video. Understanding every detail of these experiments is difficult as the collection methods and analytical techniques for each respective method are the results of multidisciplinary approaches that provide deep knowledge of biology, chemistry, physics, mathematics, and computer science. However, new information can be revealed when the data is put into context of understanding what a measurement is precisely assessing from a given experiment and what factors should be considered when weighing the values in combination in the efforts to generate any given model.

2.3.1 Structural limitations

For example, while the results of x-ray crystallography produce a single structure of a myosin molecule these structures in fact represent the average bias of thousands of molecules (Sweeney and Houdusse 2010). In addition, formation of myosin crystals is extremely difficult and numerous modifications have to be performed on the myosin (i.e. reductive methylation, removal of the heterogeneity in light chain species and varying levels of phosphorylation) to obtain high quality crystals (Rayment et al. 1993). Furthermore, an actin-bound myosin structure is yet to be resolved. Considering the strong evidence that the cross-bridge cycle is actin activated (i.e. accelerated powerstroke and P_i -release rates in the presence of actin (Lynn 1974)), this is an important limitation to be considered (and perhaps the biggest). Lastly, myosin's kinetics are influenced by many factors including load, pH, ionic strength, and $[P_i]$. It is not currently possible to variably apply loads (besides those arise from crystal packing forces) to myosin in x-ray crystallography or electron-microscopy and the solution buffers often have fixed pH and salt concentrations that optimize the formation of crystals and that are not necessarily preferential for myosin's kinetics.

2.3.2 Functional limitations & the “unifying” hypothesis

Now, the functional assays are not without faults of their own. To start, muscle fiber studies, ATPase, stopped-flow, and FRET are all bulk assays. These assays are measuring the average response of thousands of molecules of myosin into a single measurement of the development in tension, powerstroke rate, or P_i -release rate. Even optical trapping measurements of the powerstroke rate from a single molecule of myosin is not as straightforward as it might sound (ask me how I know) and involves an ensemble averaging technique (Blackwell et al. 2021; M. Capitanio et al. 2006; Sellers and Veigel 2010; Veigel et al. 1999, 2003). Moreover, in the trap single molecule displacements are the summation of the distance produced by myosin’s powerstroke and that of brownian capture (additional distance caused by thermal noise) and the transition rate going from a pre- to post-powerstroke (i.e. unbound to bound) in the trap is limited by the viscous drag forces imposed by the solution onto the trapping beads (König 2000; Svoboda and Block 1994). As a result, the rate of the powerstroke is often underestimated due to imperfect alignment of single molecule events in the ensemble averages and also in part to the transition rate between unbound/bound periods being similar to the trapping system’s corner frequency. In the case of the coupled assays where attempts are made to simultaneously measure the rate of force development in fibers while also recording the P_i -release rate, assessments must be made about the relative time resolution between the two coupled measurements to assure there is a fair comparison between the two respective rates. To this regard comes one of the biggest flaws within the powerstroke first argument. The most current argument put forward by structural biologists who support a P_i -gating model claim that the P_i -release rate measured by muscle fibers and with stopped flow is simply the wrong rate. When Llinas et al. (2015) provided new structural evidence of the P_iR state, they also hypothesized the structural sequence of P_i being released: 1) Pre-powerstroke state (P_i in active site), 2) P_iR2 (P_i transitions from

active site into putative P_i -release tunnel, and 3) P_iR1 (P_i at exit of P_i -release tunnel in solution). A phosphate binding protein binds P_i in solution with the equivalent structural state being P_iR1 or later. Moreover, the structural biologists argue that the P_i transitioning from the active site in the pre-powerstroke state to the P_iR2 state is what actually gates the powerstroke - not the transition into the P_iR1 state - and that the PPS- P_iR2 transition would occur very rapidly. This was proposed to provide a unifying theory between the structural and functional assays that bridges the gap between the powerstroke first and P_i -gating models. *However, this hypothesis remains to be rigorously tested and provides a testable hypothesis - If the transition of P_i from the active site into the P_iR2 position gates the powerstroke, then can the powerstroke be prevented by occupying the active site with P_i ?*

2.4 The S217A mutation in myosin V

Several key amino acids in the active site have been identified to interact with the ATP and more specifically, the γ -phosphate while in the active site (Forgacs et al. 2009; Gulick et al. 2000; Llinas et al. 2015; C. A. Smith and Rayment 1996). Moreover, previous simulations of P_i leaving the active site via the different proposed escape routes highlight important contacts between certain amino acid residues with this fleeting P_i along each proposed route (Cecchini, Alexeev, and Karplus 2010; Reubold et al. 2003). By manipulating the charge or size of these amino acids along the different escape routes with point mutations several research groups have tested whether they could slow the measured release of P_i with a stopped flow assay (Forgacs et al. 2009; Llinas et al. 2015). Serine 217 (S217) was shown to make contact with the γ -phosphate in the active site from its location on SWI and the S217A mutation drastically reduced the P_i -release rate ~10-fold (Forgacs et al. 2009). With this mutation, S217 was identified as playing an important role in P_i -release. Llinas et al. (2015) verified that the S217A mutation drastically slows P_i -release

and in addition showed S217 mediates the transition of the P_i from the active site into the “ P_i -release tunnel” via the back door mechanism with x-ray crystallography. Furthermore, the underlying mechanism of the S217A mutation that slows P_i -release was suggested to be due to the mutation preventing the P_i from transitioning into the P_i -release tunnel via the back door escape route, possible due to a steric blocking by a water molecule or loss of interaction that “guides” the P_i into the release tunnel (Forgacs et al. 2009). Comparatively, the E146A mutation proposed to contact the -phosphate via the alternative side door mechanism (via a SWI movement) had no effect on the P_i release rate (Cecchini, Alexeev, and Karplus 2010; Llinas et al. 2015) which was an observations that led to them supporting the back door mechanism which is currently the prevailing P_i -release pathway hypothesis (Robert-Paganin et al. 2020). Most importantly the S217A mutation provides the perfect test of the “unifying” theory proposed by the structural biologists. The unifying theory states that the transition from the active site into the P_i -release tunnel gates the powerstroke (not the release into solution). With the S217A mutation, we have a mutation that prevents the release of P_i from the active site into the P_i -release tunnel. So, if the release of P_i from the active site into the escape tunnel does gate the powerstroke then the powerstroke should be delayed or prevented with the S217A mutation in myosin V. Furthermore, since the P_i -release rate becomes so much slower ($\sim 30/s$) the observation of a delayed stroke would be obvious in a standard three-bead laser trap assay which has millisecond resolution. Additionally, myosin V has a much slower and rate limiting ADP-release rate which results in an increased attachment time when compared to fast skeletal myosin II making the attachment events in the laser trap more easily identifiable at high ATP concentrations.

2.5 Analysis of single molecule trap data

“...the interpretation of data from such experiments [the laser trap] is not straightforward” - Guilford et al. 1997

The laser trap (or optical tweezers) has been revolutionary to the myosin world. Originally developed by Arthur Ashkin of Bell Laboratories (Ashkin et al. 1986) the laser trap was adopted by biologists to study the interactions of a single molecule of myosin with a single actin filament. After the first single molecule three-bead assay was performed in 1994 by Finer, Simmons, and Spudich (Finer, Simmons, and Spudich 1994), single molecule myosin biophysics began to flourish as other research groups began building their own laser traps to investigate the basic mechanical and kinetic properties of the myosin family. While quick to adopt the method of the three-bead assay, the analysis of single molecule laser trap data seemed to be controversial even since the original Finer, Simmons, and Spudich (1994) experiment. Most of the major research groups that were trapping with myosin in the 1990s seemed to have their own opinions on how to best extract the necessary information from their raw data records and several analysis methods seemed popular while others were abandoned or seemingly ostracized by the community. Even the original work of Finer, Simmons, and Spudich (1994) came under scrutiny relatively quickly by the newly found field since they analyzed their data “by eye”, manually selecting where acto-myosin interactions occurred in the data. Moreover, even more recently as analysis has moved more hands-off and has become increasingly more automated with computer programs. Unfortunately, the details to recreate these programs are not always provided with enough details in publications and there is currently no GUI software that is easily accessible to perform the analysis of laser trap data. This makes reproducible research quite difficult, and not just from the point of wanting to replicate someone else’s experiments, but also since many reviewers of journal manuscripts expect to see laser trap data analyzed just as “someone else” has done. The lack of software is a large gap in the field and ultimately provides a hindrance to the field progressing as a whole. Following is a review, attempted at being chronological as best as possible, of the (mostly early) trapping papers and their techniques used to

analysis single molecule myosin data from the three-bead laser trap. The purpose is to review what has and is currently being done to analyze single molecule trapping data in hopes highlight the “best practices” that should be included (or would be desired) in a modern software package.

2.5.1 Manual identification

The most simple and easiest way to analyze laser trap data is just to look at it. Manual event identification (the “by-eye” technique) was the original analysis method used by Finer, Simmons, and Spudich (1994). Here, they defined several criteria that would define their “event population” (actomyosin interactions) and would scan through the data record manually marking where they believed each event would start and end. The criteria they used were that 1) events needed to be isolated events with baseline returning to approximately the same “zero” or reference position on each side of the event, 2) the displacements could not have multiple interactions (what most people now called “runs” in mini-ensemble data), and 3) displacements smaller than or equal to the variance of the baseline would not be included.

The reason analysis techniques can be controversial in the field is because many times the information provided by the data and what conclusions can be drawn are in part limited by the analysis performed. For instance, major motivation of the original three-bead assay was to provide a direct measurement of single molecule myosin displacements since there was a large discrepancy from displacement estimations provided by bulk *in-vitro* assays (motility and fibers). Additionally, the first myosin crystal structure had just been solved a year prior by Rayment et al. (1993) which had put structural constraints on myosin’s displacement to be ~6nm. However, Finer, Simmons, and Spudich (1994) directly measured a powerstroke that was about twice as large. Why? Their analysis technique was flawed. While Finer, Simmons, and Spudich should no doubt be applauded for their pioneering work, the manu-

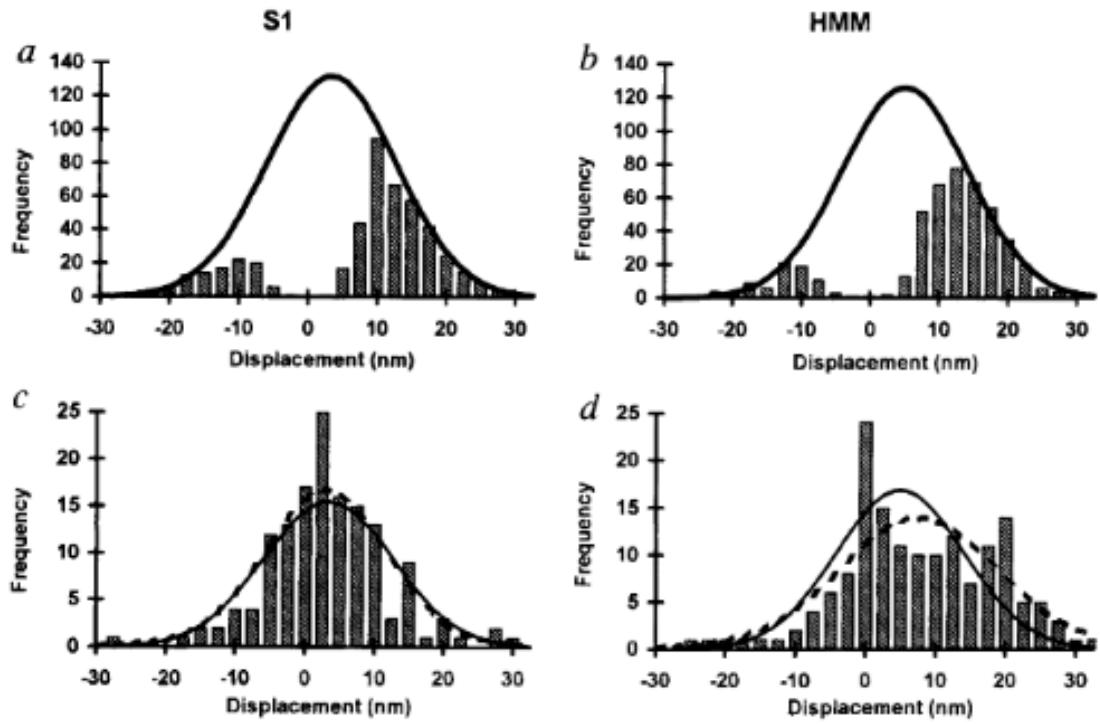
ally identification of events caused an over-estimation of their estimate of myosin's displacement (and to be fair hindsight is 20:20 and you have to start somewhere). Ultimately, what occurred is that their inclusion criteria that they defined specified events that needed to exceed the level of the baseline noise of the trapping system, or to be greater in magnitude than the variance of the baseline. Consequently, the analysis then only included the largest displacement events that occurred effectively filtering out the smaller displacements. We now know single molecule displacement events should be normally distributed with a mean equal to the average displacement and variance equal to that of the baseline signal (J. E. Molloy et al. 1995b), Finer, Simmons, and Spudich (1994) essentially truncated their distribution and inflated their mean displacement value. So yes, analysis is important.

Even so, the analysis of trap data continued to hinder the field in their ability to accurately measure the displacement caused by myosin's lever arm that matched those estimation provided by structural biologists. Two years later Molloy et al. 1995 (Biophysics) measured S1 and HMM displacements to be \sim 15nm. The analysis was similar to Finer et al. 1994, yet J. E. Molloy et al. (1995a) seemed more definitive about setting displacement and time ($>10\text{ms}$) thresholds for the inclusion of events. This method has become commonplace for use with mini-ensemble data where events must be greater than a given displacement threshold for a minimum duration to be considered into the "event population". Interestingly, Molloy et al. 1995 (Biophysics) admits in their own paper that "longer displacements could arise because both their [Finer 1994] and our data events were analyzed by eye; this may have introduced observer bias."

2.5.2 Variance threshold

Another Molloy et al. 1995 (Nature) paper was subsequently released with a new set of trapping data that was analyzed with a variance threshold (the first instance of

a variance based analysis). Previous trapping papers solely used the raw displacement record to identify attachment events, but the introduction of the variance threshold was clever and forms of this are still in wide use today in analysis programs. The principle behind the variance threshold is theoretically simple and practically robust. A small bead trapped by a laser will provide a “noisy” baseline signal that is characterized by Brownian motion. On our trap in the Debold lab, trapping at a laser power of 1.5 Watts usually produces an approximate laser trap stiffness of 0.04 pN/nm. This stiffness value (0.04 pN/nm) is far less than the stiffness of a myosin head. Consequently, as the bead-actin-bead assembly is pushed around by Brownian forces the dumbbell can travel up to ~40nm in either direction (~80nm total excursion distance). However, since a myosin head is far stiffer than trapping laser, when myosin binds to actin there is a large reduction in the variance in the signal and the bead experiences less overall excursion. While, comparing the variance of the unbound to bound population can hard to resolve by eye transforming the original raw displacement records into running variance records makes the difference in bound/unbound variance become more readily apparent as this variance signal-to-noise ratio can exceed a 2:1 ratio (usually at least >4:1 if you are Chris Marang). With the variance threshold analysis J. E. Molloy et al. (1995b) were able to identify zero and low displacement events that would have been excluded from previous analyses. Including zero and low displacement events produced a more complete data distribution to perform summary statistics on which provided a more accurate representation of the true underlying mean that was being estimated from the sample. Ultimately, they were able to conclude that previous measurements of the powerstroke displacement were overestimated due to flawed analysis which produced an incomplete distribution to be collected and biased the final average.



While great strides were made with the variance threshold implementation of J. E. Molloy et al. (1995b) a few issues have persisted since. For one, a clear rationale for the criteria that is used to define a threshold was never really provided leaving one to think this may be arbitrarily set and left to the analyzers discretion as they seek to optimize event identification “by eye” through manipulation of the threshold. Also, no methodological details are provided about how this analysis was performed in regards to software or programming languages. Both of these cause major issues when attempting to reproduce the analysis they argue as superior. Furthermore, this paper becomes heavily cited throughout future research papers conducted by these same authors that end in high impact journals setting the precedent for what is expected from the analysis of laser trap data. Veigel et al. (2002) is in *Nature Cell Biology* (impact factor ~20 now in the year 2021), has been cited 425 times, and only provides a minimal explanation of the analysis performed and instead simply opts to cite J. E. Molloy et al. (1995b) as reference for their analysis methods which also has an air of ambiguity around the analysis methods.

2.5.3 Correlated thermal diffusion

The first few years of myosin laser trapping mostly used one quadrant photodiode (QPD) to track the position of one of the trapped beads of the bead-actin-bead dumbbell. Mehta, Finer, and Spudich (1997) implemented a two QPD setup where they simultaneously could track the position of both beads. With this additional data they implemented an analysis method they called “correlated thermal diffusion”. Since a bead-actin-bead assemble is a rigid system when the setup is unbound from myosin the beads move in unison. As in, the position of one bead is correlated with the other one as they are pushed around by Brownian forces (linked together by a pre-tensioned actin filament). However, when the stiffer myosin head interacts with the actin filament the beads become “disconnected” (figuratively not literally...though, I guess, literally disconnecting is possible, but a separate issue...just use a lot of biotin and then you do not really have to worry about any literal disconnection). As a result, the position of the two trapped beads becomes uncorrelated. Practically, the correlated thermal diffusion analysis can be performed by iteratively applying linear regressions over small windows (data subsets) of the raw displacement data from the two traps and returning the correlation coefficient of the fit. The “event population” will be the transient periods in time when the correlation coefficient drops to zero. The main advantage of correlated thermal diffusion is that the variance signal-to-noise ratio does not have to be as high as in the variance threshold analysis which is usually dependent on the amount of pre-tension that can be applied to the bead-actin-bead assembly and the “stickiness” of the beads. This analysis is quite similar to the covariance analysis method that also utilizes the advantages of having two QPDs, but instead calculates the covariance of the two beads position instead of extracting correlation coefficients from linear regression fits. Since covariance analysis requires two QPDs and our setup only has one, it is not a viable analysis option and will not be included in this review in detail.

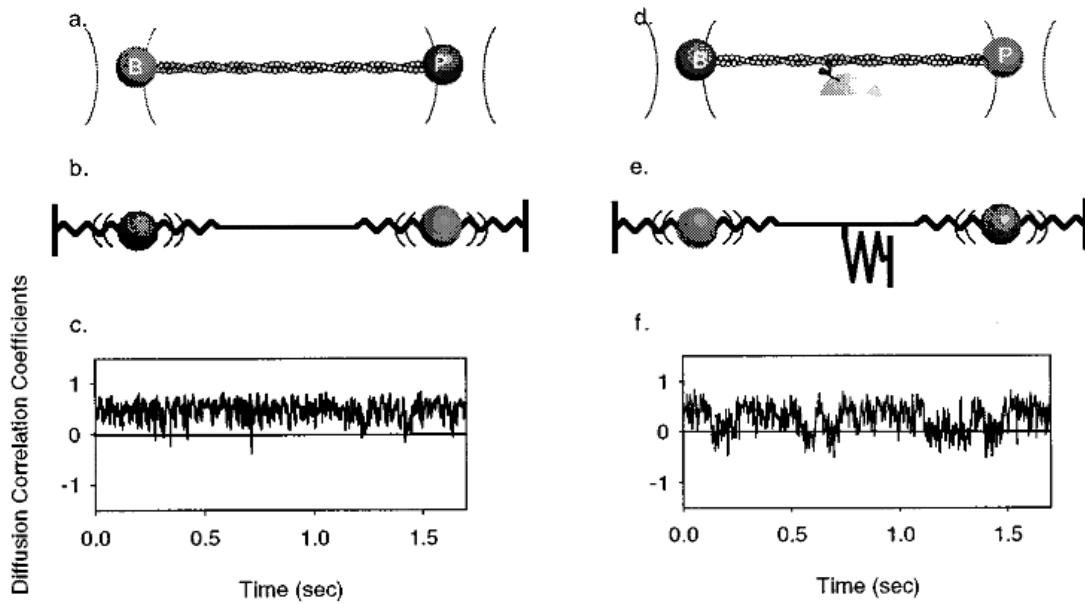


Figure 2.5. Diagrammatic explanation of the correlated thermal diffusion analysis from Mehta et al. 1997.

2.5.4 Mean Variance Analysis

While the original motivation of using the single molecule laser trap was to measure the displacement a single myosin molecule could generate, the first few years produced a discrepancy of displacement estimations that were largely analysis based due to the “eye-balling” of arbitrarily thresholded data. Guilford et al. (1997) (Warshaw Lab at the University of Vermont) re-purposed an analysis technique originally used for single ion channel data to perform the analysis of their single molecule trapping data called “mean variance analysis”, or MV (Patlak (1993), both research groups are from the University of Vermont). This technique is a completely different approach than any of the previous attempts at analyzing laser trap data (with myosin as the motor) and extracting displacement data from. Implementation of MV in the laser trap was in response to the previous attempts of estimating myosin’s displacement which were made by manually selecting binding events from thresholded data (Finer, Simmons, and Spudich 1994; J. E. Molloy et al. 1995a), a practice that was sub-

sequently shown to be not of “best-practice” as it was not statistically defenseable (Block and Svoboda 1995). The benefits of MV is that it provides a model independent transformation of the data, requires no manual selection/scoring of events by eye, and requires no assumptions about the underlying data. However, the procedure of performing analysis is more complex.

Mean-variance analysis is performed, and so aptly named, by transforming raw trapping data (displacement vs. time) into mean-variance histograms. The MV histograms are 3-dimensional histograms that are constructed by calculating the mean and variance of a “window” (small subset of data). While, the direct results of calculating these windows values results in two time series data set, these can be plotted against each other to construct a 3-dimensional histogram where mean is on the x-axis, variance is on the y-axis, and the “third” z-axis is counts/number of windows/data points and reflects the total time spent at any given mean-variance combination. Since the baseline data (myosin unattached from actin) is characterized by Brownian motion the result of a histogram has a prominent normally distributed “baseline” population with mean centered around zero. If any actomyosin interactions are in the data traces they will present as an alternate “event” population with mean value equal to myosin’s displacement size. Additionally, the “event” population will have a smaller variance than the baseline due to myosin being stiffer than the trap. The “event” and “baseline” population can be separated by their variance, typically a data trace that contain no actomyosin binding event is used to “remove” the baseline from the data with events. After, the removal of the “baseline” population the “event” population can be fit by a sum of gaussians to estimate myosin’s displacement.

Furthermore, Patlak (1993) showed that average attachment times and the number of events could be estimated with the MV approach. In MV, since the “third” (z-axis) represents the counts/amount of time spent at a given mean-variance this “time” is dependent on the window width used to construct the MV histogram (smaller win-

dows width produce a data set with more data points). By iteratively constructing MV histograms and recording the number of “counts” in the event population, a plot of counts vs window width can be constructed and this relationship is defined by a single exponential. Fitting the exponential $V_{mv} = t_{on}ke^{(N-1)t_{on}}$, provides estimates of the average attachment time (t_{on}) and number of events (k) present in the data which are parameters of the fit.

While thresholding and identifying events “by-eye” is simple, it is also (relatively) easy requiring no sophisticated software or advanced computer programming skills. MV has benefits as an alternative analysis, but it is more computationally intensive and seemingly impossible to perform without a computer and advanced programming skills (iteration, curve fitting, statistical F-tests proving the merit of additional Gaussian, and automation). To the credit of the authors, both Patlack and Guilford, there exists a GUI based program that allows a user to perform MV analysis. However, the program runs on Windows only, is not open source, is not actively maintained, and not obtainable without a connection to UVM. Limitations of MV analytically for trapping data is that each “event” population is biased by the length of an individual event and not the number of events itself. For example, if a data trace has 10 events, nine of them 4nm and 10ms long, and one being 20nm and 100ms long, the average of that “event” population will be more biased toward the value of the longer event than it would if the mean was calculated mathematically. As a result, ideally you need a lot of data (events) to produce a full distribution. Additionally, in MV there still is a more or less arbitrary threshold that is needed to separate populations if subtraction of the baseline is not used. However, the biggest drawback of MV is that the analysis does not actually identify individual events, but populations. This limits the ability to perform any post-analysis procedures on the individual event (ensemble averaging) that has become commonplace (and almost expected) from a modern trapping paper. Nonetheless, this method is still viable even today in the

modern trapping world as our lab as had recent success using MV with a 2020 paper detailing how a non-nucleoside substrate differential affects myosin function in a laser trap (Woodward et al. 2020).

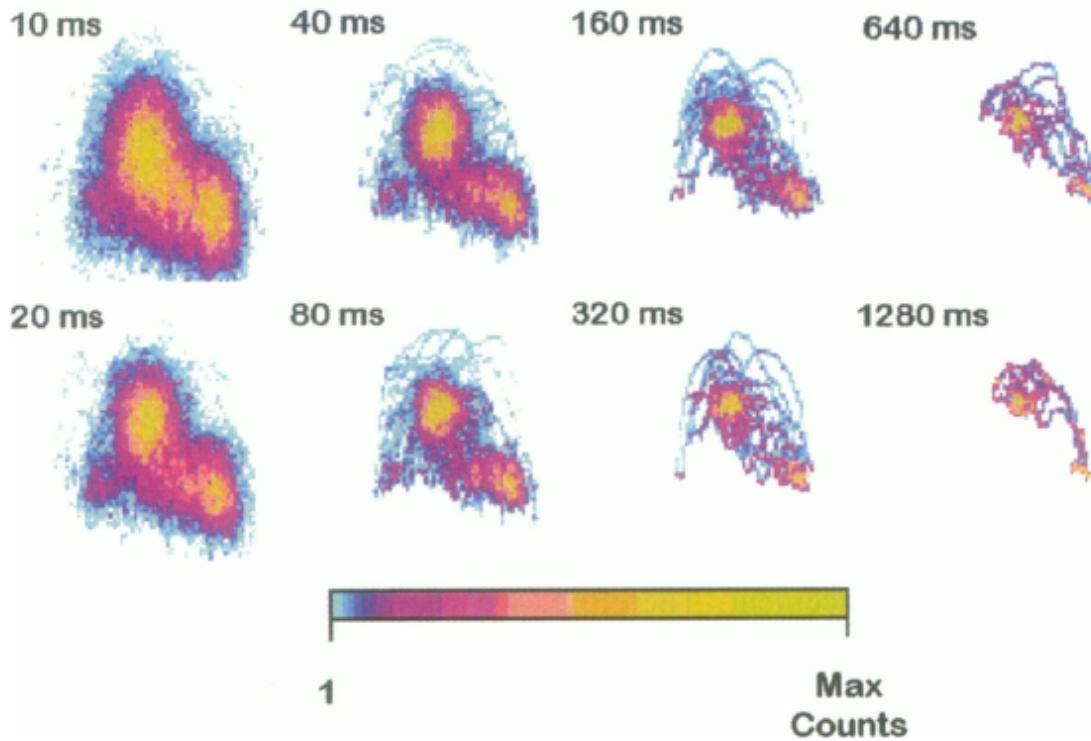


Figure 2.6. MV histograms constructed at different window width from Guilford et al. 1997

2.5.5 Page Method

For most of the 90s the previously reviewed methods, mainly thresholding and MV, was the mainstay for single molecule myosin laser trapping. Seemingly, there seemed to be large controversy over the superiority of the two methods as inferred from the review by Knight 2001. This article is a part analysis review/part introduction of the Page Method as a technique for analyzing trap data (note Veigel and Molloy are co-authors). In this review, not only is MV written-off, but the authors' (UVM crew) trapping skill seemingly outright insulted. In response to the method

Guilford et al. (1997) used to remove the baseline population to avoid arbitrarily setting a variance threshold to define an event population, Knight et al. (2001) remarks “Fig. 3c shows that the two populations can be clearly distinguished without recourse to such measures if the experimental data is of sufficiently high quality”. Bold! The Page Method is an analysis that provides automatic event identification and is a re-vitalization of an older analysis technique that was originally described as a “continuous inspection scheme” by E.S. Page (PAGE 1954). In short, this method relies on using probability distribution functions (PDFs) in attempts to classify data points (of unknown origin) as a part of a certain populations, or distributions, based on comparing the probability of obtaining a data point with that specific value from a set of given PDFs in order to detect a change/switching of the underlying PDF. Knight et al. (2001) describe how to perform a version of the Page Method for the analysis of laser trapping data. First, a running mean can be subtracted from the original trapping records. The result is a removal of all displacements from the data so the “baseline” and “event” population should both be characterized as normal distributions centered around zero. However, the two populations will differ in their respective variances. The baseline will have a larger variance as dictated by Brownian motion and the event population will have a smaller variance since the increased stiffness of the myosin head dampens the effects of Brownian motion on the trapped bead’s position. These variance levels can be calculated directly from the data or inferred. Since a mean and variance level can be estimated for the two populations, baseline and event, PDFs can be constructed and the probability of achieving each data point from each of the two distributions can be calculated. The log odds ratio is then expressed for each data point comparison. Due to the shapes of the two population (baseline is much wider due to higher variance) points nearest zero have a higher probability of being from the event population. This produces positive odds ratios, whereas anything with a higher probability of belonging to baseline has a neg-

ative log odds ratio. Since positive values indicate attachment events and negative values un-attachment, a zero threshold cumulative sum can be calculated over all of the resulting log odds ratios to identify the start and end of all events. Since the attachment events return positive log ratios the cumulative sum will steadily increase throughout the attachment time of a true (real) event. When the event ends negative logs odd ratio will result and the cumulative sum will return back to zero leaving a peak in the cumulative sum trace identifying the end of an event. Performing the same analysis in reverse will identify the starts of the events. There is a minimum duration threshold that needs to be set under which peaks in the cumulative sum should be ignored to reduce detection of false events. Interestingly, while this paper was submitted from a well-known trapping group and paints the Page Method in a positive light this method does not appear often or in subsequent use of analysis in their future work (it was not used in Veigel et al. (2002) - Nature Cell Biology). For what anecdotal evidence is worth, I had a lab-mate discourage me from using the Page Method as it was conveyed to me that this method was not reliably capturing shorter attachment events. Perhaps this analysis is best suited for lower ATP concentrations and/or slower motors.

2.5.6 Hidden Markov Model

In response to the controversy surrounding both the previously described analysis techniques and the inconsistency in reports of myosin's actual powerstroke size D. A. Smith et al. (2001) proposes the Variance Hidden Markov Model (HM-Model) as an alternate analysis technique that unlike the other analyses actually provides a “best-fit” to the data by removing arbitrarily set thresholds in exchange for using a kinetic model to describe the underlying probability of being in a given “state” (bound/unbound). The main goal of a HM-Model is to predict the occurrence of some un-observable event from another (related) observation. In relation the the

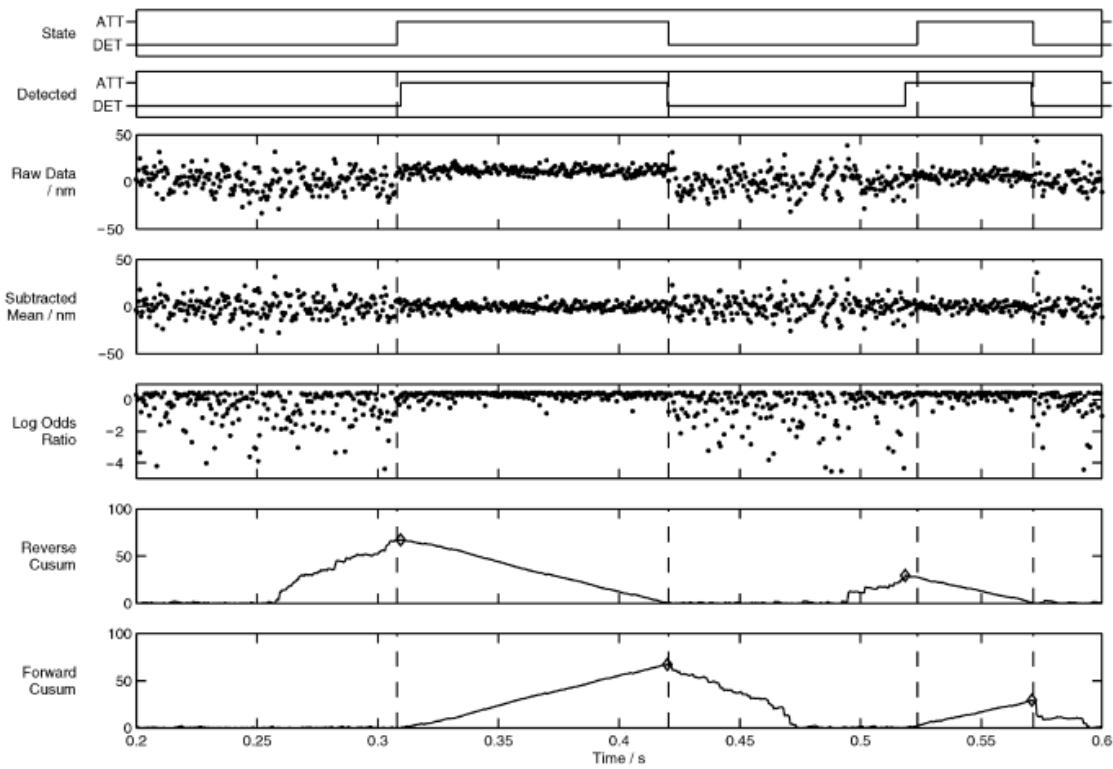


Figure 2.7. Event identification via the Page Method from Knight 2001

laser trap, the objective is to identify periods of time when myosin is bound to actin. However, you cannot actually physically see myosin bind to actin in a laser trap. This is in fact an un-observable event. But, we do know that myosin is stiffer than the laser traps, so if myosin does bind the variance of the displacement signal should decrease. This is the information in the data that is collected and that can readily be observed (bead position over time). With the HM-Model, the goal is to use the statistical characteristics of the two variance populations (unbound/bound) to predict the most probable sequence of un-observable states throughout the data. The parameters in the model include myosin's attachment and detachment rate (commonly denoted as f and g , respectively), and the variance of the unbound/bound populations. Ultimately, the probability for each variance window belonging to a certain "attachment state" can be determined by comparing the probability for each window to belong to a certain variance PDF in combination with factoring in what the previous attachment state was. In a HM-Model, a certain variance window would not necessarily be categorized as belonging to an "attachment state" just because there is a higher probability of the window to be drawn from the bound variance PDF. The HM-Model also considers the "transition probability" by taking into account the previous attachment state. The transition probability in this case is myosin's attachment/detachment rate. So, the categorization needs to determine the most likely attachment state for the current variance window based upon the conditional probability of observing a certain variance value given the prior attachment state. For an un-elegant example and if the algorithm had human thoughts perhaps it would be asking these questions to-itself while it tried to determine the attachment state given the previous state was unattached: "What's the chance of a myosin binding event occurring and then what is the chance of drawing this variance value from the attached state variance PDF?" versus "What's the chance of the myosin staying unattached and then what is the chance of observing this variance value from the unattached

variance PDF?”. Though you *could* perform the attachment state de-coding by hand (it really is just multiplying probabilities and selecting the highest one to determine the state), this is performed computationally via the Viterbi algorithm. Furthermore, the parameters for the model can be estimated using the Expectation-Maximization (EM) algorithm. So a practical use case involves first estimating the model parameters with the EM-Algorithm and then using the Viterbi algorithm to perform the state sequence decoding. The HM-Model thus provides a sophisticated and robust approach to analyzing single molecule laser trap data as this technique has a simple assumption of a basic kinetic scheme for myosin dynamically binding and unbinding from an actin filament without arbitrarily defining thresholds to identify events. Furthermore, this approach was implemented when M. Capitanio et al. (2006) observed the hitch in skeletal muscle myosin II for the first time showcasing the HM-Model’s ability to reliable detect true events. However, this is a more advanced approach that would require a lot more effort and knowledge (as compared to a user defined threshold) of both math and computer programming to implement and automate from scratch. Unfortunately, there are no computer programs available to easily perform this analysis on laser trap data which is probably why it is not in high spread use (even though it arguably should be).

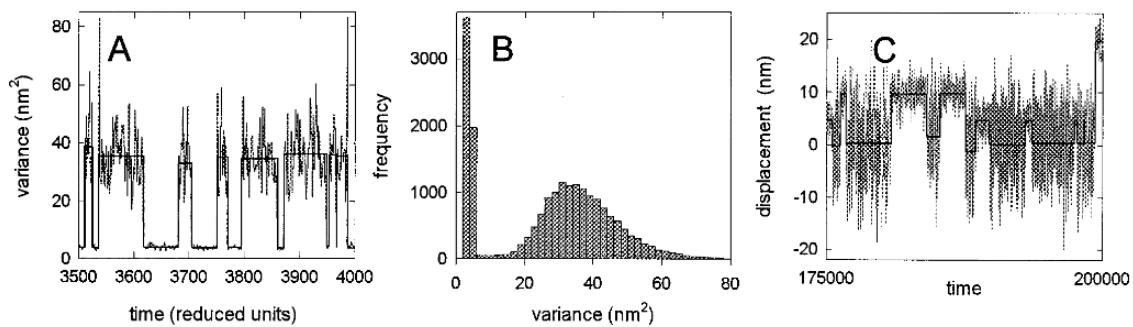


Figure 2.8. Hidden Markov Model identifies single molecule events from Smith et al. 2001

2.5.7 Ensemble Averaging

One of the advantages of using an analysis that identifies actomyosin binding events is that these methods provide information about when the event begin and ends (aka these analyses report at which data point an event start and at which data point the events end). This is quite useful information to have. For one, being informed when events are starting/ending allow the user to be able to visually inspect the data and the resulting analysis to decide whether the analysis appears to be identifying “true” events or not which allows the user to be more confident in their data if they are visually pleased with the results. But also, having an analysis program that identifies individual events allows additional post-analyses to be performed. A common such test for laser trapping data is ensemble averaging. To ensemble average trapping data first you need data that has been analyzed with some sort of event identification analysis. Once all the data has been analyzed and events identified there is a couple unusual transformations that occur. Furthermore, a complete ensemble average consists of both a forward and backwards ensemble average which differ by the initial alignment of the events. Forwards ensemble averages are aligned at the start of the events and the backwards ensembles are aligned at the end. To conduct an ensemble average, first all events are subset out of the original data trace and are extended to the length of the longest event. Then the events can be aligned horizontally by being placed on the same relative time scale and each point averaged vertically so the first data point of the first event is averaged with the first data point of the “nth” event. The results of the ensemble average is “one event” that represents the average response of all the events. By performing this analysis additional information can be estimated from trapping data. For instance, with the forward ensemble averages the rate of the first and second powerstroke can be estimated; however, providing estimates for the rate of the first powerstroke is rare as the rate of the first powerstroke is far greater than the time resolution of a standard rate. Some

researchers ignore this rate, but this rate can still be fit and used to represent the rate at which the unbound to bound transition occurred for a given condition with the caveat being the rate is not reflective of the rate of the actual powerstroke, but one that would reflect the powerstroke and the resulting movement of beads through solution. Ensemble averaging has been used extensively to measure both the size and rate of the second powerstroke, sometimes called “the hitch”. Additionally, since the hitch is often associated with the release of ADP many use the rate of the second powerstroke as a surrogate measure of the ADP release rate that can be estimated from trapping data. The size of the hitch can be estimated from a floating parameter to the exponential fit that estimates the rate, or by subtracting the beginning position of the backwards ensemble from the final displacement from the forward ensemble. Lastly, the second order binding constant of ATP to myosin can be estimated from an exponential fit to the backwards ensemble. As a result, ensemble averaging serves as a powerful post-analysis tool which can provide important kinetic and mechanical insights to the originally collected data set.

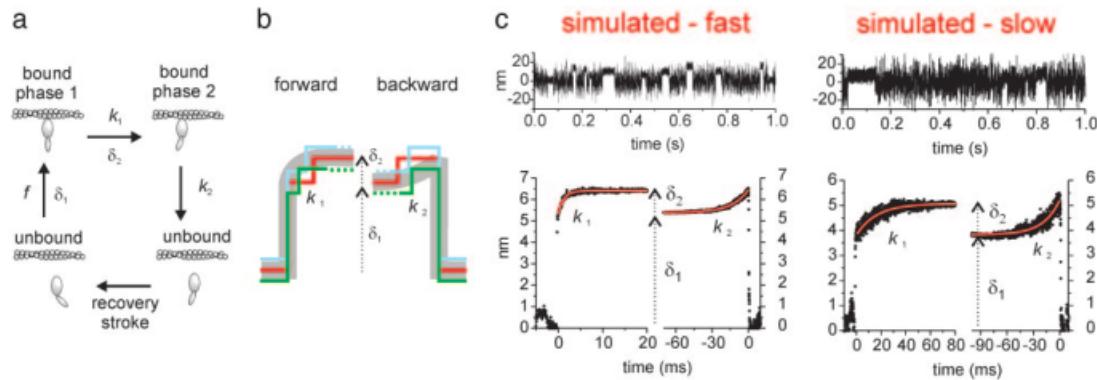


Figure 2.9. Ensemble averaging data from Capitanio 2006.

2.6 Ultra-fast force clamp (UFFC)

In almost all of the previously described analyses techniques in Chapter 2.6 the variance of the trapping signal is used to somehow differentiate between myosin being either attached or un-attached from actin. While this is a reliable method in regards to detecting single molecule events, the time resolution afforded by the standard laser trap setup of two beads stuck in stationary traps is limited in which these analysis method can be used. As a result the use of the signal variance for event detection can only be pushed so far. In a standard laser trap setup, the time resolution is ultimately limited when using a variance driven analysis method due to the roll-off frequency of the baseline signal of the trap, dictated by brownian motion and the stiffness of the traps:

$$f_c = (1/2\pi) * (kx/6\pi\eta r)$$

Unfortunately, the limited time resolution is not a simple issue related to the analysis technique used or even to the sampling frequency. The problem with the limited time resolution in a standard laser trap setup using an event identification program (or when applying real-time feedback) based on changes in system stiffness/variance is that there is a lack of signal in the necessary frequency range (Knight et al. 2001). This means that changes in variance are only detectable for frequencies far less than the corner frequency of the trapping signal (Marco Capitanio et al. 2012) leaving the shortest detectable events to be ~5-15ms in a typical trapping setup (Knight et al. 2001; Marco Capitanio et al. 2012) as temporal resolution is inversely proportional to the roll-off frequency (Neuman and Nagy 2008) and most research groups report roll-offs between 300-500 Hz (Veigel et al. 1998; Guilford et al. 1997). In theory, the temporal resolution can be increased under standard trapping conditions by increasing system stiffness or by decreasing the drag on the beads (Neuman and Nagy 2008). But in practice, increasing baseline stiffness would then decrease the signal-to-noise ratio that is typically used to detect binding in three-bead assay. To combat the lim-

ited temporal resolution in the standard laser trap setup people started “wiggling” one of the traps. Veigel et al. (2003) was the first to “wiggle” one of their traps in a myosin based three-bead assay. The motivation of the paper was to apply load directly to a single myosin motor in order to probe the underlying mechanisms of the “Fenn Effect”. Here, the authors applied a 1kHz oscillation (sine wave) to one of the traps (high frequency, low amplitude). Attachment events were then able to be identified as deviation from the assigned amplitude to the oscillating trap. This effectively decreased their time resolution to ~1ms. Marco Capitanio et al. (2012) took this approach to another level with the introduction of the Ultra-fast force clamp (UFFC). While Veigel et al. (2003) had the time resolution to assess the load dependence of the ADP-bound state in smooth muscle myosin II at low ATP concentration, 1-ms time resolution was still too slow to directly observe or test the load dependence of the powerstroke itself. In standard optical trapping setup the “powerstroke” is not seen and the transition between unbound/bound appears instantaneous. In fact, even in the ensemble averaged data of standard optical trapping data, the “rate of the first powerstroke” is often ignored because the powerstroke rate occurs faster than the time resolution of the trap and the unbound-to-bound transition is influenced by the terminal velocity of a damped bead moving through solution. Additionally, smooth muscle myosin has a two-fold slower ATPase rate (Harris and Warshaw 1993) than skeletal muscle myosin II. Which brings up the major problem when trying to study skeletal muscle myosin II - *it is fast*. Moreover, skeletal muscle myosin II has a low duty ratio and is non-processive. This means that it spends the majority of its time off-actin and only transiently interacts with an actin filament in a stochastic manner which means it only spends a few milliseconds on actin at saturating ATP concentrations.

The UFFC has micro-second temporal resolution and sub-nanometer spatial resolution. To reiterate, the UFFC has MICRO-SECOND temporal resolution and SUB-

NANOMETER spatial resolution. If you don't think this is cool *you should probably stop reading this dissertation now*. With UFFC not only can single powerstrokes be observed, but load can be applied to the actual powerstroke as well. The increased time resolution is applied with a similar principle to what Veigel et al. (2003) performed - “wiggle” the traps. The following sentence might be the biggest understatement/oversimplification in this entire dissertation. In UFFC, both traps are “wiggled” simultaneously, very quickly. The basic setup is still a three-bead assay, but once a dumbbell setup is formed, both traps are rapidly displaced back and forth which results in a triangular wave being imposed on the setup. The side-to-side displacements are performed by first putting equal and opposite forces on both of the beads (via AOD control), and then additional force is added to one of the traps which steers the setup in a certain direction at a specified force level. Once, a pre-configured distance is reached ($\sim 200\text{nm}$), the additional load is then revoked from the one trap and added to the other in order to switch the direction of the setup. Switching directions of the movement allows the actin filament to remain within binding distance to the myosin on the mogul. One advantage of UFFC is that the dumbbell setup is always experiencing the load, so when a myosin does bind, the load will be instantaneously (well, near-instantaneous...this time depends on the stiffness) applied to the myosin as there is no waiting period for a feedback system to engage once an actomyosin interaction is detected. Events are detected by converting the raw positional data (bead displacement over time) into a velocity measure and applying a threshold. The velocity of the dumbbell ends up being equivalent to the total force applied to the system divided by the viscous drag of the dumbbell setup $v = F_{total}/drag$. Since the UFFC is designed to apply a constant force, when myosin binds the traps will rapidly stop moving in order to maintain the desired force level. This results in the velocity transformation of the raw UFFC displacement record forming a double Gaussian with the bound population being centered on zero (nm/ms). Currently, only two UFFC

experiments have been performed with myosin. The original UFFC experiment was performed with skeletal muscle myosin II Woody et al. (2019). Woody et al. (2019) performed UFFC with cardiac myosin II under both 0mM and 10mM-P_i. As previously described, the UFFC provides the perfect test of the main proposed question of this dissertation - “What is the relative timing between the powerstroke and P_i-release?” because UFFC allows for direct observation of the powerstroke and this can be tested under both 0mM and 30mM-P_i concentrations. This technique allows to test the question without a mutation that slows myosin’s kinetics (i.e. “making a slower mouse”) because it has the necessary time resolution to directly observe if the presence of high P_i levels, which increase the probability of myosin re-binding to the active site, can prevent and/or delay the powerstroke (i.e. UFFC is the “better mouse trap”). Indeed, Woody et al. (2019) observed that the presence of 10mM P_i did not alter the rate of the powerstroke in cardiac myosin. Thus concluding that the powerstroke precedes the release of P_i from the active site; however, structural biologists have disputed these claims as mis-interpretation of the results of their UFFC data and actually see Woody et al. (2019) as provided evidence that P_i-release gates the powerstroke (Robert-Paganin et al. (2020)). Clearly, there is still a need to perform UFFC with high P_i, especially with skeletal muscle myosin II since it has yet to be done.

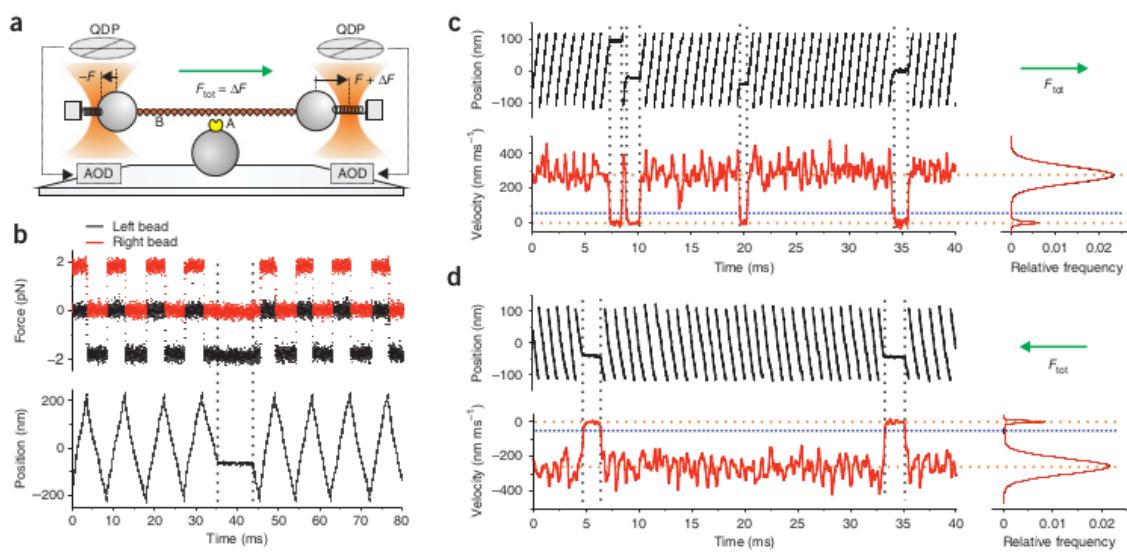


Figure 2.10. The Ultra-Fast Force Clamp from Capitanio 2012

CHAPTER 3

METHODS

3.1 Protein Isolation and Expression

3.1.1 Myosin V expression

Myosin V S1 is expressed (baculovirus system) with the first 792 amino acid residues which includes 1 IQ domain as detailed previously by the Yengo Lab (Gunther et al. 2020). Additionally, expressed myosin V contains the N-terminal tetracysteine motif, C-terminal Myc, and C-terminal FLAG tags (Trivedi et al. 2013; Trivedi et al. 2015, 2020; Gunther et al. 2019). The S217A mutation was introduced (serine to alanine) using QuikChange site-directed mutagenesis (Stratagene), co-expressed with calmodulin, and purified with FLAG affinity chromatography.

3.1.2 Skeletal muscle myosin II isolation

Fast skeletal muscle myosin II was isolated from chicken pectoralis muscle (Demand Farm, Wendell, MA). All procedures during isolation were carried out in a cold room or performed on ice. Isolation was performed as previously described by the Debold lab (Woodward et al. 2020; Unger and Debold 2019; Longyear, Walcott, and Debold 2017) by Mike with a protocol similar to those of Margossian and Lowey (1982) with minor modifications. In short, chicken breast muscle is passed through a meat grinder and rinsed with 0.2M EDTA. 2 mL of Buffer A (Extraction buffer consisting of 0.3M KCl, 0.15M KP_i, 20mM EDTA, 5mM MgCl₂, 3.3mM ATP, and 5mM DTT at pH 6.7) is added per gram of tissue including 5mL of protease inhibitors. This is mixed for 12 minutes with an overhead stirrer. The reaction is stopped with

a 4X dilution into water which is then mixed and filtered. After precipitate settles it is centrifuged at 10,800g for 10 minutes at 4C and the resulting pellet is resuspended with Buffer B (Suspension buffer consisting of 1M KCl, 60mM KP_i, 20mM EDTA, and 5mM DTT at pH 6.7) and mixed gently before being left to dialyze overnight. Actomyosin is then precipitated and centrifuged at 41,171g for 1 hour at 4C and the resulting supernatant is diluted 10X with water. Clear supernatant is siphoned off and the rest is centrifuged again at 10,800g for 15 minutes at 4C. Supernatant is poured off and precipitate is resuspended with Buffer D (Resuspension buffer consisting of 3M KCl, 50mM KP_i, and 5mM DTT at pH 6.7) before being dialized overnight for a second time against Buffer E (Dialysis buffer consisting of 0.6M KCl, 50mM KP_i, 1mM NaN₃, and 5mM DTT at pH 7.0). After the dialysis, myosin is clarified with an ultracentrifugation at for 2 hours at 4C, concentration determined, snap frozen with liquid nitrogen, and stored at -80C.

3.1.3 Actin isolation and labeling

Acetone powder was prepped from the remainder from the myosin isolation (with the leftovers from the filtered cheesecloth) and actin purification was performed from the resulting acetone powder as described by Pardee and Spudich (1982) with modifications. Briefly, acetone powder was finely ground and mixed with extraction buffer (2mM Tris Base, 0.2mM CaCl₂, and 0.005% NaN₃ at pH 8.0) and stirred with an effort to minimize creation of bubbles. The resulting solution is spun at 28960g for 20 minutes and supernatant filtered off and kept aside. Additional extraction buffer added to gel-like precipitate and centrifuged a second time with the same specs with the supernatant filtered off and combined with the previous. Actin is polymerized from the resulting supernatant by addition a final polymerization solution (50mM KCl, 2mM MgCl₂, and 1mM ATP). Salt is added to slowly to prevent “salt shocking” the proteins before being left to stir overnight. The next day, a high salt wash

(increase KCl to 600mM) removes tropomyosin from the f-actin and then the sedimentation of f-actin performed by centrifuging at 205835g for 60 minutes. The precipitate is transferred to a homogenizer and resuspended with extraction buffer. A 4 day dialysis is performed with extraction buffer additionally containing ATP and DTT to de-polymerize actin. After dialysis the resulting G-actin is clarified with an ultracentrifugation at 200,000g for 60 minutes. After actin is polymerized by adding 10mM Imidazole (pH 7.0) and 1mM MgCl₂ and dialyzed against final storage buffer (4mM Imidazole, 25mM KCl, 2mM MgCl₂, 1mM NaN₃, and 0.01mM ATP at pH 7.0). After calculation of final concentration actin is snap frozen in liquid nitrogen and stored at -80C. After isolation actin can then be labeled with 100% TRITC for use in vitro motility or mixed with a 50:50 TRITC/Biotin solution for use in the three-bead laser trap assay.

3.2 Laser trap assay

The laser trap assay was performed as previously described by the Debold Lab (Woodward et al. 2020; Unger and Debold 2019; Longyear, Walcott, and Debold 2017) with special considerations for the expressed myosin V. Single molecules of myosin were adhered to a nitrocellulose coated microscope slide containing 3 μ M glass pedestal beads with an additional coverslip glued on top for construction of a “flow-cell”. The final myosin concentration of ~0.8-1 μ g/mL was added after introduction of anti-myC antibody (0.8 μ g/mL, Sigma Inc.) which provided a binding interface for the expressed myosin on the surface. Bovine Serum Albumin (BSA) was used to block the remainder of the surface before the addition of final buffer. The final buffer consisted of an actin buffer (91mM KCl, 1mM EGTA, 4mM MgCl₂, and 1mM DTT at pH 7.0) mixed with 100 μ M ATP, and an oxygen scavenger system (29mM glucose, 1.5mM glucose oxidase, and 80 units catalase) at pH 7.0. For 30mM P_i experiments KCl was reduced in order to maintain the 125mM total ionic strength to match the

control 0mM P_i experiments. The concentration of TRITC/Biotin labelled actin filaments and neutravidin/streptavidin coated 1 micron beads (Bangs Lab Inc) was varied at trappers discretion. Bead-actin-bead “dumbbell” setups were constructed using a three axis piezo controlled stage (Mad City Labs) with a time shared laser trap between two positions. Experiments were performed at 1.5 Watts laser power and actin filaments pretension to 3-4pN. The resulting system stiffness of the two laser traps and the pretension across the filaments was 0.04pN/nm, determined via the equipartition method (DUPUIS et al. 1997). Bead position was tracked using a four quadrant photodiode with a sampling rate of 5kHz.

3.2.1 Analysis of laser trap data

All of chapter 4 is dedicated to this topic.

3.3 Ultra-fast force clamp

The UFFC data was collected under the same experimental conditions and buffers as described in the section 3.2 for the standard laser trap assay, but full length fast skeletal muscle myosin IIx was adhered to the nitrocellulose coated coverslips instead of the myosin V and anti-body combination. Since UFFC is an extension of the standard three-bead assay implementation was similar in regards to setting up a dumbbell. However, in order to increase the speed that the dumbbell was able to move through solution when the force clamp was engaged smaller 510nm beads (Bangs Lab) were coated with neutravidin (31000, Pierce) for use in the UFFC. Decreasing the radius of the beads reduces the Stoke’s drag coefficient since $\beta = 6\pi\eta r$, where β represents the drag coefficient, η the viscosity of solution, and r is the radius of the bead. Furthermore, the drag coefficient is inversely proportional to the velocity of the dumbbell setup as given by $velocity = F_{total}/\beta$. The smaller the drag the faster the dumbbell can move. Consequently, the increased velocity of the dumbbell setup

in the un-attached state increases the time resolution and signal-to-noise ratio of the resulting data. Force was pre-determined for each experimental condition and set by calculating the bead's displacement from the center of the trap and converted to a force since $F = -kx$, where x is the bead's displacement and k is the stiffness of the trap. The trap stiffness was similar to the standard laser trap experiments at roughly 0.04 pN/nm calculated by analysis of the power spectrum. Positions of the traps were controlled through the AODs and a custom LabView program. Bead position was collected at 200kHz sampling frequency. The total feedback delay was around 8;micro:s, this is the total time it takes for the computers to detect changes in the beads position relative to the center of the trap, communicate the information, and for the AODs to respond to the changes.

Actomyosin interactions in UFFC are identified by applying a threshold the velocity transformation of the raw displacement data that is generated during data collection. The velocity was calculated instantaneously on a point-to-point basis and the results smooth with a Gaussian filter. The velocity transformation results in a double Gaussian distribution with two peaks corresponding to the average velocity of the bound and unbound populations. The bound population velocity is centered around zero because as myosin binds actin and imposes its own stiffness/drag the UFFC will feedback in attempts to apply a consistent force to the dumbbell causing the traps to stop moving. The threshold is set at the point along the joint PDF of the double Gaussian where the probability of crossing the threshold due to noise from the unbound or bound event is equivalent. The threshold was then optimized for each record to decrease the number of false events detected to <1%. If false events exceeded 1%, the SD of the Gaussian filter was increased to further smooth out the data in order to decrease the probability of a baseline noise artifact crossing the threshold. Note that usually smaller SD of the gaussian filters could be applied at great forces due to an increase in force subsequently increasing the signal-to-noise ratio since the

baseline velocity is faster to achieve higher forces. Additionally, there is a correction factor that is applied to determine the start and end of the events that is a result of the optimal threshold being closer to the peak of the bound population in the velocity distribution that is a result of the bound population having a narrower peak (smaller SD) since myosin stiffness is greater than the trap stiffness. After event identification, events were ensembled averaged as by synchronizing events in the *x* and *y* dimensions by applying linear regressions to the baseline prior to the start of the event (when the bead is moving at constant velocity) and overlaying events at the point where the linear regression intersected the start of the event as ID'd prior in analysis. The resulting ensembles could be fit with a model consisting of a linear portion that described the delay before the powerstroke and a double exponential reflecting the kinetics/mechanics of myosin going through the first and second powerstroke. Ensemble averages are typically calculated separately for the three populations of event durations that are prevalent in UFFC experiments (short, intermediate, and long) since these events represent different mechanochemical schemes of an actomyosin interaction.

CHAPTER 4

LASERTRAPR: A COMPUTATIONAL TOOL FOR AUTOMATING THE ANALYSIS OF LASER TRAP DATA

4.1 Introduction

The laser trap (or optical tweezers) has been revolutionary to the field of single molecule biophysics. Originally developed by Arthur Ashkin of Bell Laboratories (Ashkin 1986) the laser trap was eventually adopted by biologists to study the interactions of single molecular motors (e.g. myosin, kinesin, dynein) with their molecular tracks (e.g. actin, microtubules) by use of a three-bead assay (Finer, Simmons, and Spudich 1994; Kojima et al. 1997). These experiments permit researchers the ability to observe the interaction of two proteins within a millisecond-time and nanometer-spatial resolution providing unprecedented insight into the molecular machinery underlying a wide variety of biological functions including muscle contraction, intracellular cargo transport, and cell-division. Such experiments need to be performed with a low trap stiffness (0.02-0.04 pN/nm) as to not hinder the function or harm the integrity of the experimental proteins or setup. Since the position of a trapped bead is largely dominated by Brownian forces, a bead stuck in a trap with low stiffness has a large variance in its displacement signal as trap stiffness (α_{trap}) is inversely proportional to the variance (σ^2) of the displacement signal (via the Equipartition Theorem) Svoboda and Block (1994).

$$\alpha_{trap} = \frac{k_B T_k}{\sigma^2}$$

where k_B is the Boltzmann constant and T_k the temperature in Kelvin. The high variance of the baseline displacement signal combined with the dampening effects of viscous drag forces masks the underlying mechanics of the two proteins interacting and cycling through a mechanochemical scheme that is common amongst biological motors used in these assays which makes the analytical task of identifying these events of interests quite challenging.

The variance of the displacement signal is a crucially important feature of single molecule laser trap data as the variance can be exploited to determine when protein interactions do occur. Since the biological motors used in these experiments are stiffer than the trapping laser, the interaction of the proteins can be characterized by a decrease in signal variance of the time series (position over time) signal, via Eq 1, which is also often accompanied by a displacement from the mean baseline position as in the case of a biological motor, like myosin, attaching to an actin filament and performing a powerstroke. In some cases, the signal-to-noise ratio of the baseline and event populations variance can exceed 2:1 which makes these interaction events readily discernable “by-eye”. However, while simple and easy, the analysis of data “by-eye” has been criticized in the past as this method was suggested to introduce subjectivity via user bias as evidenced by early inaccurate estimations of myosin’s displacement size (Finer, Simmons, and Spudich 1994; J. E. Molloy et al. 1995b). This exemplifies the fact that while the laser trap is a powerful and advanced scientific instrument, the reliability and accuracy of the information that can be extracted from the resulting data is limited by the validity of the techniques and programs used to analyze the data. While there are numerous techniques that can be used to identify binding events, a common theme between them is that most require advanced computer programming knowledge to implement. This is then compounded with a need to then automate those scripts by a preferential creation of user-friendly graphical user interfaces (GUIs). For most, the advanced computer skills required

to build sophisticated analysis programs with GUIs are taught in classes that are not degree requirements for graduate students or researchers seeking degrees in many biology-related fields. This presents a technological barrier that hinders progress in understanding and interpretation of data for new students and researchers, and an additional monetary cost barrier is added to a laboratory if the creation of custom program must be outsourced. And, even in these situations then a research group is left with an custom and un-supported “black-box” program.

Unfortunately, there are currently no completely open-source projects whose primary aim is to automate the workflow of analyzing laser trap data (calibrations, processing, event identification, ensemble average, and summarizing statistics) written with an open-sourced programming language. Although it should be noted there have been recent publications of programs aimed at single molecule event identification, most notably the MATLAB based SPASM (Software for Precise Analysis of Single Molecules, Blackwell et al. (2021)). However, while SPASM itself is an open-source program, the underlying MATLAB language is proprietary/closed source language and has a steep financial barrier (currently a standard MATLAB license has an annual fee of \$860 per their website at the time of writing). Here, we present **{lasertrapr}**, and open-source program for automating the analysis of laser trap data written in R, a free and open-source programming language (hence `lasertrapr` = laser trap + R; also note that it is common in the R-community to denote R-packages with `{}`). The tool has an easy-to-use GUI provided by the R-Shiny web-framework package. One of the main benefits of having a tool built with R/Shiny (R Core Team 2022; Chang et al. 2021) is that there is high portability of the app across different operating systems as it can be installed on Windows, MacOS, and Linux systems. Additionally, we do not view our application as a replacement or competitor to a program such as SPASM, but as an additional tool made available to the biophysics community that has some similar features (single molecule event identification and ensemble averaging) but for

distinct data types (SPASM’s main event identification is built for 2 QPD systems using co-variance of bead position whereas our is for a 1 QPD system). Furthermore, lasertrapr fully embodies the notion of a free and open-source project whose main goal is automation and reproducibility of the entire workflow of analyzing laser trap data which includes folder/file creation and organization, signal calibrations, data cleaning and preparation, event identification of both single molecule and mini-ensemble data, ensemble averaging, generation of complete project summary statistics, and creation of publication quality figures. Lastly, the co-existence of multiple programs will only benefit the biophysics community by enabling researchers the ability to contribute to and use an analysis program best suited for their interests and experimental setups.

4.2 Results & Discussion

The following Results & Discussion serves as both a validation of the app and provides example use cases of what can be accomplished within the app. The paper presented in Chapter 5 was analyzed completely with this app so the present chapter’s aim is to provide evidence that the app is doing what we think it is doing. If you cannot trust the analysis you cannot trust any of the results or interpretations that are based off of the data produced from it. While reading this section if you decide that you *really* like the app and would like to try it yourself, there is a user-guide and complete documentation on how to install and use the app available on the app’s website <https://lasertrapr.app/> including example videos. The documentaiton is also included in section 4.3. Additionally, the project can be found on Github at <https://github.com/brentscott93/lasertrapr>.

4.2.1 Single Molecule Analysis Validation: Simple

One of the main features of the app pertains to the analysis of single molecule laser trap data. Here, we will validate the single molecule analyzer that is based on a

combination Hidden-Markov Model + Changepoint analysis. The full details of the analysis are provided in section 4.2.7 which walks through the actual single molecule analysis line-by-line. These tests will verify that the app can identify actomyosin binding events using simulated data which provide an ideal and known conditions. The first set of simulations are **simple**. In fact, the entire dataset is the exact same baseline-event-baseline motif concatenated together 200 times. Every event is the same - 5nm displacement, 100ms attachment time. The single baseline-event-baseline data set is shown in Figure 4.1.

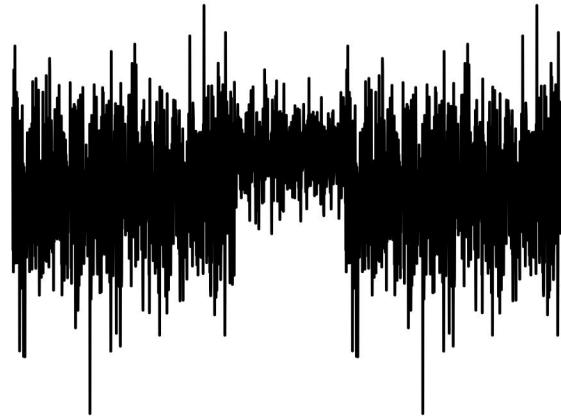


Figure 4.1. A simulated laser trap event. The event has 200ms of baseline preceding a 5nm displacement that lasts for 100ms. Another 200ms of baseline data is appended after the displacement. This exact sequence was replicated and concatenated together 200 times which yields a data trace with 200 simulated binding events that are spaced 400ms a part. The exact simulated measurements are: 5nm displacement, 100ms attachment times, and 400ms time between events.

After creation of the 200-event “simple dataset”, we analyzed the trace to see how many events were detectable out of the original 200 created. One of the features of the app is that after the analysis it takes the original raw/simulated trapping data and “overlays” the analysis results onto the original data so the user can compare/check how the analyzer is performing. Of course, the app auto-saves the files of all the relevant data files automatically for the user to a pre-determined file-folder hierarchy completely in control by the app (no fussing with files/folder anymore!).

Please refer to the user manual for more information on folder/file management and details about the data that is being saved. Although an important note is that the data is all saved in plain text files (.csv/.txt). All data is transparent and easy for the user to take/import/manipulate into other software as they choose. The user's data is **their** data. However, as a cautionary note, manually manipulating *lasertrapr* folders and files is highly discouraged except for highly advanced users, since the app can easily be broken by removing/altering the contents of files/folders it generates. A screenshot of the analysis results as viewed from within the app for the simulated "simple dataset" is shown in Figure 4.2. Within the app the figure displayed in Figure 4.2 is actually an interactive graph that allows users to pan across their data and the analysis allowing the user to become more familiar with their data and the app's output.

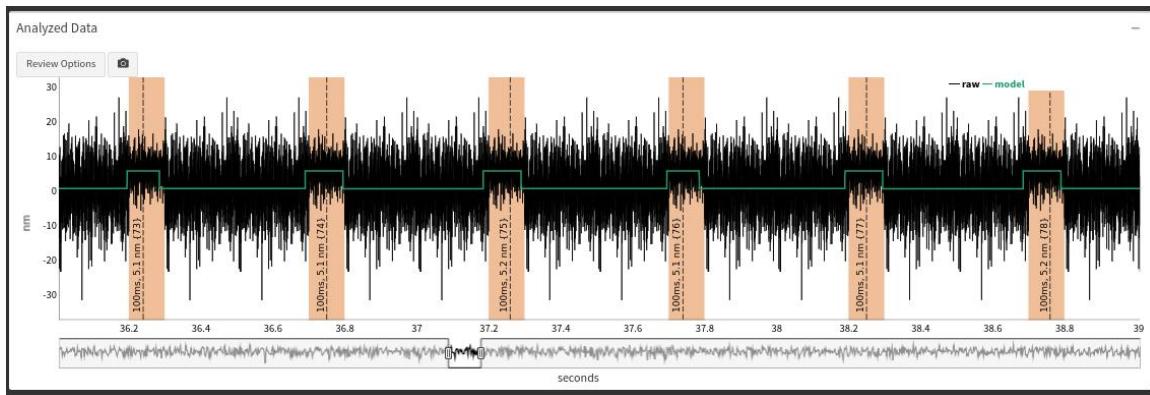


Figure 4.2. Screenshot of the output from the single molecule analysis produced by *lasertrapr*. Black lines is the original trap trace, the green line signifies the results of the hidden markov model performed on the running window transformation, and the yellow highlights is the final results after changepoint is applied.

The output of the simple simulation test is shown in Table 4.1. The results show that the analysis correctly identifies and measures single molecule binding events. The app estimated a 5.1nm displacement and 99.8ms attachment time which accurately represents the known simulated values of 5nm and 100ms. Indeed, the 400ms time between events is also the same as the simulation input.

Table 4.1. App correctly identified the displacement size, attachment times, and time between events

displacement_avg	time_on_avg	time_off_avg
5.109922	99.8	400.0251

While this is seemingly a simply problem, with an un-realistically simple dataset, it was perhaps the biggest and hardest hurdle I faced upon starting to build the first version of the single molecule analyzer years ago.

4.2.2 Single Molecule Analysis Validation: Adding Complexity

4.2.2.1 Simulating displacement distributions

Adding complexity to the simulations creates more realistic datasets that allows for more robust testing and assurance in the precision and accuracy of the program. In the previous “simple” data example, every event had the same exact displacement and attachment time. Here we simulate displacement distributions which more accurately represents how real single molecule data would be collected. Since each displacement in the trap is the summation of the displacement caused by myosin’s powerstroke and that of random brownian noise, displacement data in the single molecule laser trap have been shown to be normally distributed with a mean displacement equivalent to myosin’s powerstroke size and standard deviation that is dependent on the trap stiffness (J. E. Molloy et al. (1995b)).

Four datasets were simulated whose event population had displacements generated from four distinct Gaussian distributions whose mean values were 0, 5, 10, and 15nm, respectively. Being able to detect 0nm displacement events was an important test to conduct considering the possibility for the “slow mouse” S217A mutation to actually have a small or even no displacement as described in *Chapter 1* within Aim 2. The data was analyzed with the single molecule analyzer (HMM+Changepoint) and summarized in Figure 4.3. The app’s estimated average displacement was modeled

against the actual average displacement of the Gaussian distributions that generated the data and fit with a linear regression. The coefficient of determination (R^2) is 0.99 indicating that the analysis is able to accurately predict average displacements across a wide range of distances. Additionally, out of a total of 4000 simulated events the app identified 3997 which is a 99.925% detection rate for these simulations.

4.2.2.2 Simulating attachment time distributions

In similar fashion to the displacement distributions, the attachment lifetimes for each myosin binding event is obviously not the exact same for each interaction. The total event population for myosin's attachment time is exponentially distributed, so we simulated additional data where we defined exponential distributions to generate total attachment times from for each event. Truncated exponential distributions were used in order to more easily generate data that would be confined to the needs of the simulations. For example, in the case of a “regular” exponential distribution and as the PDF suggests, the smallest numbers are the most probable to generate upon random sampling. However, when simulating data we are confined to defining time in terms of datapoints per the sampling frequency. Moreover, the simulator is not overly sophisticated and the simulator can, at the moment, only handle a single forward pathway through myosin's cross-bridge cycle (i.e. you have to start at “A”, then proceed to “B”, and then “C”) and at each point within the simulated pathway there has to be some finite amount of time spent in that particular state. To avoid randomly generating infinitely small ADP-release/ATP-binding rates, which define the total attachment lifetimes, the exponentials are truncated at a minimum of 1 or more milliseconds. Additionally, a truncation of minimum attachment-times also increases the assurance that if we generate 100 events the simulator will actually simulate 100 *observable* and *detectable* events. For these sets of simulations, the rates supplied to the exponential distributions will represent that average attachment

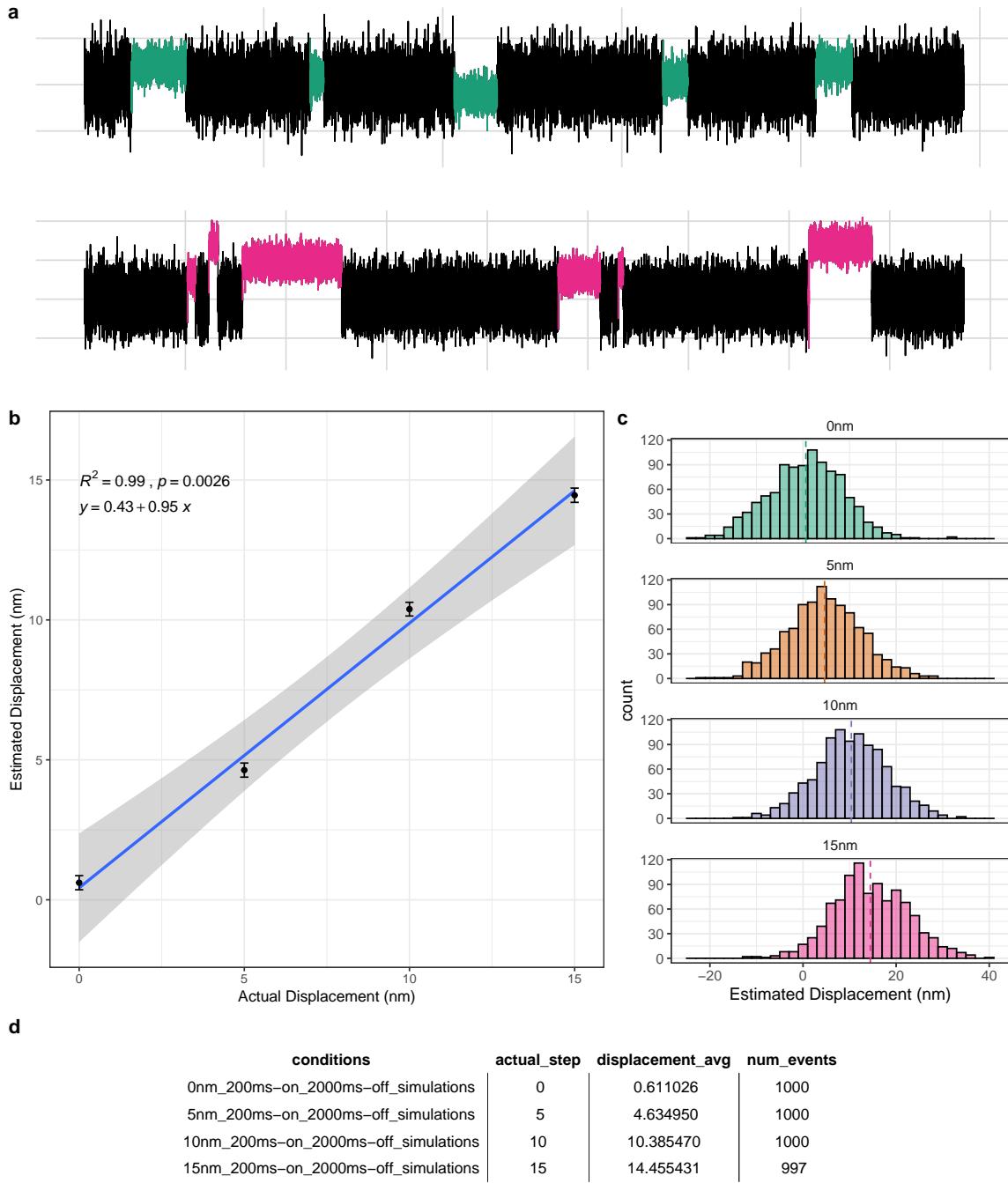


Figure 4.3. Four simulations were performed with average displacements of 0, 5, 10, and 15 nm. a) Simulated data traces. Top trace has 0 nm average displacement. Bottom has a 15 nm average displacement. b) Linear regression of the actual displacement (x-axis) versus the analysis estimated displacement size (y-axis) demonstrating a near perfect correlation. c) Full gaussian distributions of the analyzer's estimated displacement for each of the 4 simulations.

lifetime of the population since for an exponential distribution the arithmetic mean (expected value) is equivalent to the reciprocal of the decay rate

$$E[x] = 1/\lambda$$

. Drawing a random value from each distribution yields the length of time that the attachment time should be to for a given event. Each distribution shown below reflect 10,0000 random draws from a respective distribution whose rate are 20, 10, 5, 1 with lower bounds of 0.02, 0.02, 0.02, 0.02 and upper bounds of 0.3, 1.856, 1.856, 3. As shown in Figure 4.4, decreasing the rate increases increases the time values that can be generated which results in longer attachment times. Furthermore, panel C represents exponential distribution with the same rates used except panel C is **not-truncated**. The distributions in panel C further display the use-case for truncating the data to make more economical (i.e. smaller file sizes) as it is an attempt to avoid excruciatingly long attachment/detachment times when the rate constant is 1. For these simulations, the ATP binding rate was set to take 1ms (5 data points) on average, so its attachment time contributions can largely be ignored which made the comparisons easier. This would be analogous to conducting experiments at a very high ATP concentration ($>1\text{mM}$). Essentially as soon as ADP is released a new ATP is readily nearby and instantaneously can bind to myosin's active site causing a detachment.

The data was analyzed with the single molecule analyzer where the attachment lifetimes were estimated by the app and then compared to the true values set in the simulations. Since truncated exponential distributions were used, the average attachment time is not as simple as taking the reciprocal of the decay rate. To get an estimate of the possible true rate/attachment times generated from the truncated distributions, 10 rounds of 10,000 random draws from the truncated distributions were generated. For each 1 set that contained 10,000 random draws the average was

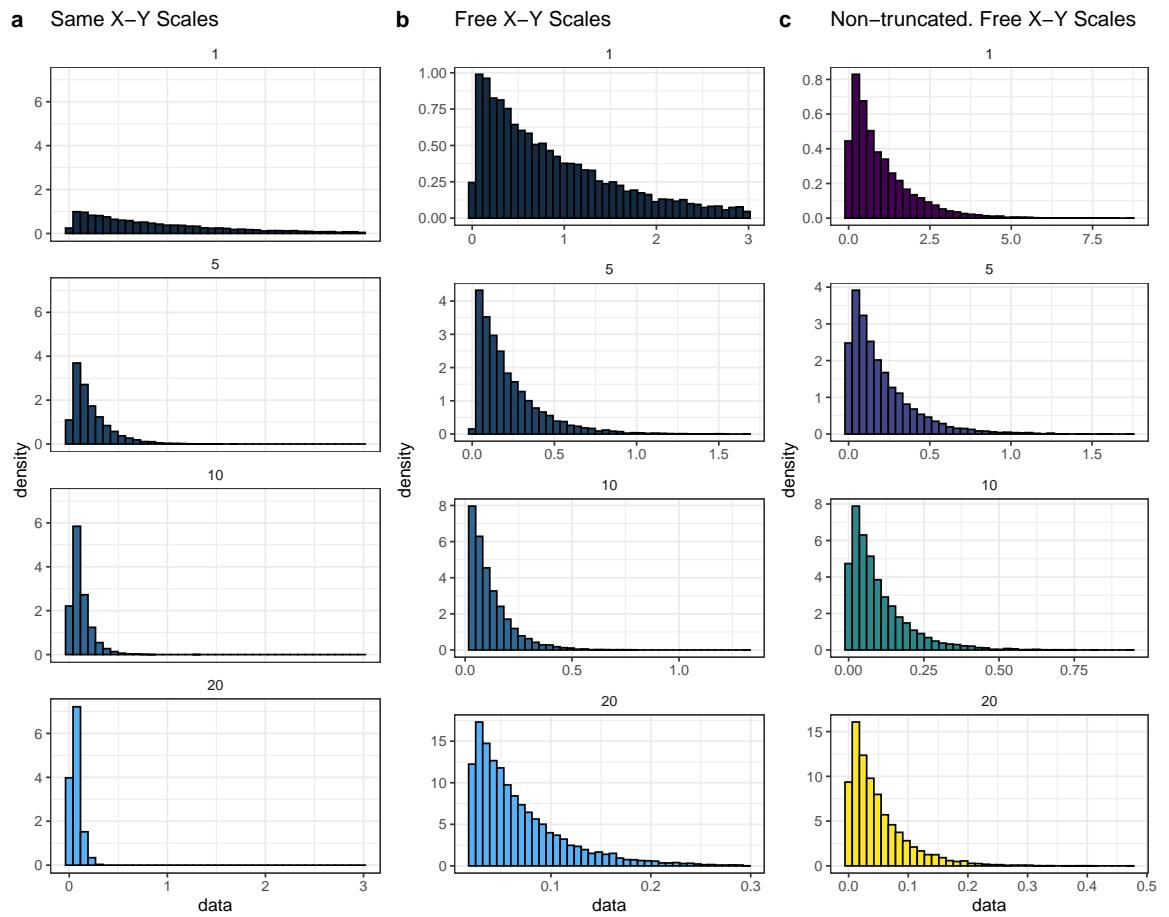


Figure 4.4. Simulated truncated (a & b) exponential distributions that are representative to the ones used to generate the simulated data

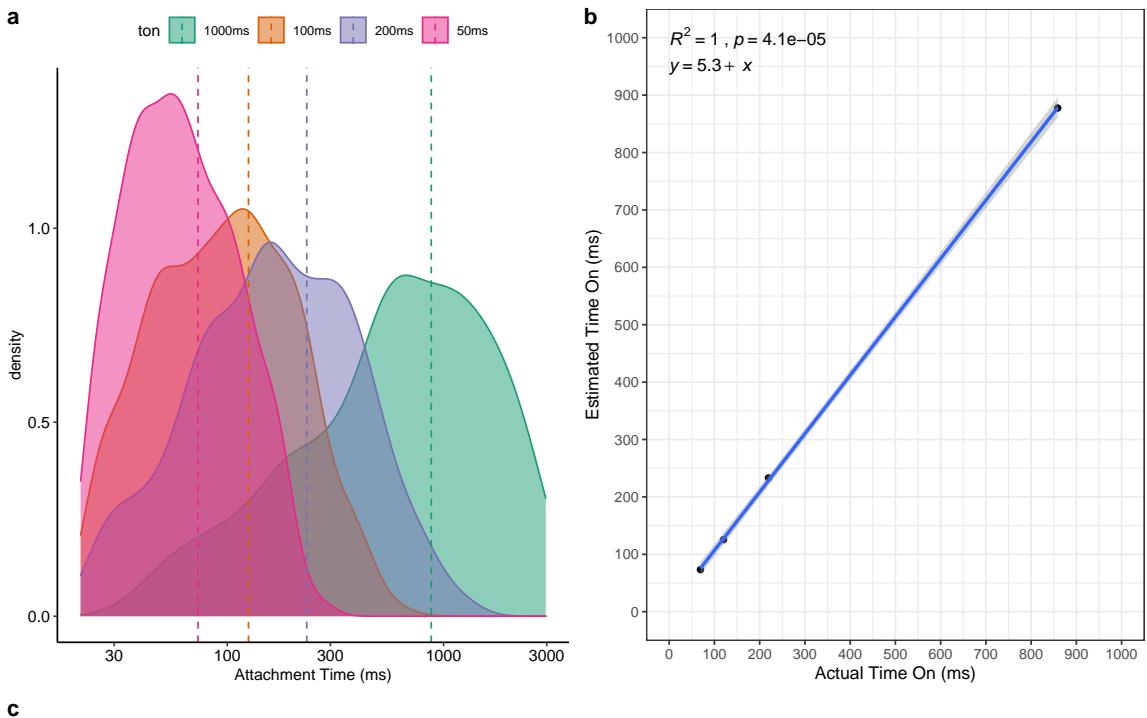
recorded (0.068759, 0.0693254, 0.0696117, 0.068123, 0.0689889, 0.0690398, 0.0698608, 0.0693016, 0.0689494, 0.0684859) and then those averages were averaged together to be compared against the mean attachment time as estimated from the app. In a similar fashion to the displacement distributions, the estimated attachment lifetimes were modeled against the known/true values, and fit with a linear regression. The coefficient of determination (R^2) is 1 indicating that the analysis can accurately estimate attachment lifetimes for a wide variety of time periods indicating that the app/analysis should be able to be applied to a wide variety of myosin/molecular motors with differing ADP release rates and even under experiments at higher ATP conditions which decreases the attachment times.

4.2.2.3 Short Events

To further test the single molecule analyzer and test the limitations of reliable event detection a dataset was simulated with very short attachment times (~20ms average). The window width of the HM-Model was decreased to 100 datapoints making the events detectable. Figure 4.5 shows the identified short events in the simulated data trace and was generated from within the app with the built-in “screenshot” tools which makes saving snippets of analyzed traces easy and provides color coding of identified free of charge.

4.2.3 Accuracy of determining beginning and end of events

The single molecule analyzer built using the HM-Model/Changepoint analysis will estimate the start or end of event by returning the exact datapoint in the original raw trace that the analyzer chooses as the most probable start/end for each event. We can compare the app’s estimation of the start/stop of each event with the true start/stop datapoint at which the event begins/ends by using a simulated data trace as the information about the exact datapoint that the event starts/ends is truly known by the simulation. Figure 4.6 shows the average/median number of datapoints that the



conditions	input_rate	true_time_on	app_time_on	num_events
1000ms_simulations	1	859	878	393
200ms_simulations	5	219	233	1000
100ms_simulations	10	120	125	1000
50ms_simulations	20	69	73	1010

Figure 4.5. Four simulations were performed with average attachment lifetimes of 50

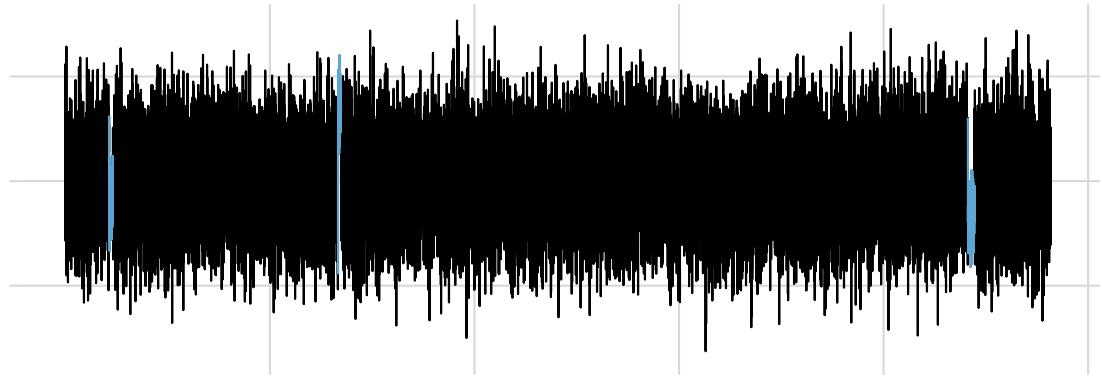


Figure 4.6. Simulated data trace with very short events. Measured attachment times for the 3 ID'd events in the trace are 18, 15, and 32ms. Plot is interactive online. Grid lines represent 1s and 2nm intervals.

estimation differs from the known values and the estimated values. Each histogram is inset with a cumulative distribution and in Panel A for the “misalignments of the start of the events” can be interpreted as ~75% events are being chosen within 2ms (10 datapoints) of the true starting datapoint. The table in panel “C” of Figure 4.6 shows the values of the misalignment calculations which were calculated by subtracting the datapoint index of the true start/end of each event (a known value from the simulations) from the app’s estimation of the start/end of the event (i.e. $\text{start}_{\text{app}} - \text{start}_{\text{true}}$). If the value of the misalignment is negative this would indicate that the app’s estimation of the start/end datapoint index tended to be a smaller number than the true value. This would mean that the app tends to start the event *early* and that the app is estimating the event to start before the event really does, by a couple of milliseconds. Looking at the comparison of the event start, the distribution appears to be more exponentially distributed so the median value, which has a value of -3, indicates that 50% of the events are misaligned within the range of -3 datapoints to the maximum misalignment +8 datapoints. Comparing this with the end of event misalignment calculations, the median value is -2 datapoints, indicating that 50% of the events are misaligned between the minimum value of -8 datapoints and this

median value of -2 datapoints. These “misalignment calculations” would indicate that the app’s estimation of the end of an event is a smaller datapoint index, meaning the app is terminating events early by about 1 millisecond. The few larger misalignment values where the app estimates the events to end after the true stop seem to be skewing the average to a positive value, but indeed for the majority of events, the app is estimating the end of each event to be slightly before the true end as evidenced by the median value and the cumulative distributions in panel “C” of Figure 4.6.

4.2.4 Ensemble Averaging

Ah, the ensemble average. This is an analysis that I am now certain that I actually did not really understand what exactly it was or even what it represented/could be interpreted (and let alone write a program to do it) for years. Conceptually this process is simple, as the final step is just a mathematical average of data points. However, the process of wrangling your data to get to the final averaging step is challenging. Ensemble averaging in general is discussed in the literature review in *Chapter 2*, and here we will stick to the discussion of ensemble averaging in regards to the app and how it implements the technique.

The main question addressed in *Chapter 5* using the “slow mouse” S217A mutation that slows P_i-release was a major motivating factor for me to start building single molecule event identification programs that would then permit the ability to perform ensemble averaging. Obviously prior to data collection we were unsure what the effects of the mutation would be, or how the severe the effects would manifest. My first analysis attempts I was using the Mean-Variance analysis as described in *Chapter 2*. However, during lab discussions the concern was raised that with MV, we would only be able to estimate a displacement distribution in attempts to see if the mutation would decrease myosin’s overall displacement. A point was made that the *total* displacement could be un-affected, but that the transition rate could still be

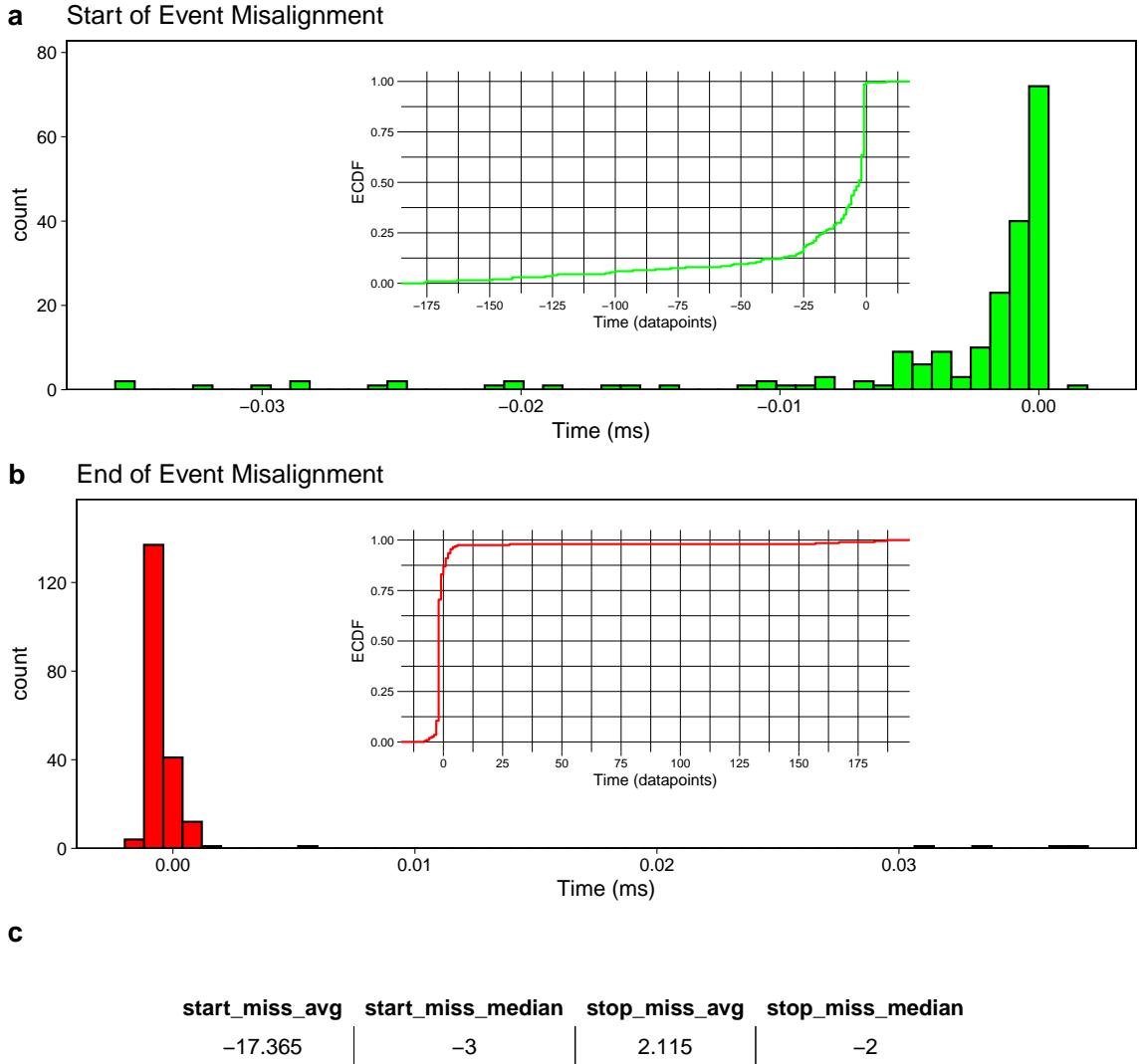


Figure 4.7. Analysis of the misalignment in the true start/end of an event and the analysis estimated start/end of an event. a) About 75% of the start of events are picked within several datapoints. b) The ends of events are mostly picked with a couple data point error. c) A table showing the average or median number of datapoints that the analysis missed the start or end of an event by.

slowed by the “slow mouse” mutation. This question was un-testable with MV since it provides a single global displacement estimate for a raw trace, so from there I set forth tje journey towards ensemble averaging.

4.2.4.1 First attempts

Developing the first ensemble averaging program was challenging as methods were generally rather limited on technical details. Part of the problem is that the process is rather simple on the surface - after automatic event detection, events are temporally synchronized, extended to the same length, and averaged vertically - done. However, finding more technical descriptions that can help navigate the development of a program to perform this analysis was less common to find details for (i.e. *how do you “extend” an event?, what specific equation/parameterization do you use to fit?, are any events filtered out before averaging?*). After many attempts my first major breakthroughs came after reading through the methods in M. Capitanio et al. (2006) which described ensemble averaging in a way that I began to understand (and I am not just saying this because M. Capitanio also just happens to be on my committee). Before diving into the current version of the ensemble average analyzer in **{lasertrapr}** we will highlight the first ensemble average that I created and used for Gunther et al. (2020) which is a pre-cursor to the experiments performed in *Chapter 5*.

Gunther et al. (2020) was a preliminary study prior to Scott-Marang 2021 where we callobrated with the Yengo & Thomas Lab doing a wide variety of biophysical characterization of the “slow mouse” S217A mutant. Chris and I preformed the laser trapping experiments (event Ned collected some data!). This was the data I started analyzing with MV and that I used to start developing the early versions of the tools now available within **{lasertrapr}**. By the way, Gunther et al. (2020) ended up receiving “Editor’s Pick” from the *Journal of Biological Chemistry* (probably because of Figure 6 and all the cool trapping stuff).

The ensemble averaging for this was slightly different from previous published ensemble averages in part because we were wanting to address *one* specific question - *does S217A slow the unbound-to-bound transition?* Some major differences were that I only included the first 60ms of all each event, only positive displacement events were included, and I only created forwards ensembles. Additionally, ensembles were fit with a single exponential that started from the origin (0,0). Most all previous attempts at ensemble averaging skip the first few milliseconds when fitting a single exponential, because 1) they are attempting to estimate the rate of ADP-release associated with the hitch, and 2) the rapid rise in the first few milliseconds is generally associated with the initial displacement, which occurs faster than the time resolution of the trap so it was seen as almost frivolous to measure the rate (understandably). However, our case was different because now we had a mutation that *could* drastically slow the transition from unbound-to-bound making the rapid rise an important feature to look into. Figure 4.7 shows the first ensembles I created before developing the app in Gunther et al. (2020) and the following is part of the methods I wrote for the paper describing the process in more detail.

“The ensemble averaged events are 60 ms long and only include events that had positive displacements. Events meeting the inclusion criteria had the back 30% of their lengths removed in preparation for event alignment. The changepoint analysis provided a new relative time index for the start of each event, and as a result the events could universally be reindexed with the same relative time scale. The first data point in each event was indexed as dp0, and thus each event was indexed as {dp0, dp1, . . . , dpn}, where dpn represents either the 300th data point (dp299) in the event (corresponding to the first 60 ms of a longer event) or the last data point of a shorter event (dp,299). For short events less than 60 ms in which dpn , dp299, the average displacement of the event was used to extend the event by dp299 - 2 dpn data points to make all events equal lengths. Events were then aligned horizontally at dp0,

and all matching dpx values were averaged together to create the final ensemble. The average ensembles were then fit with a negative mono-exponential equation using the drc and aomisc R packages to provide estimates of the rate of the working stroke and plot with the ggplot2 and cowplot R packages. Additional programming tools used for building the analysis programs include the gtools, pracma, and tidyverse packages.”

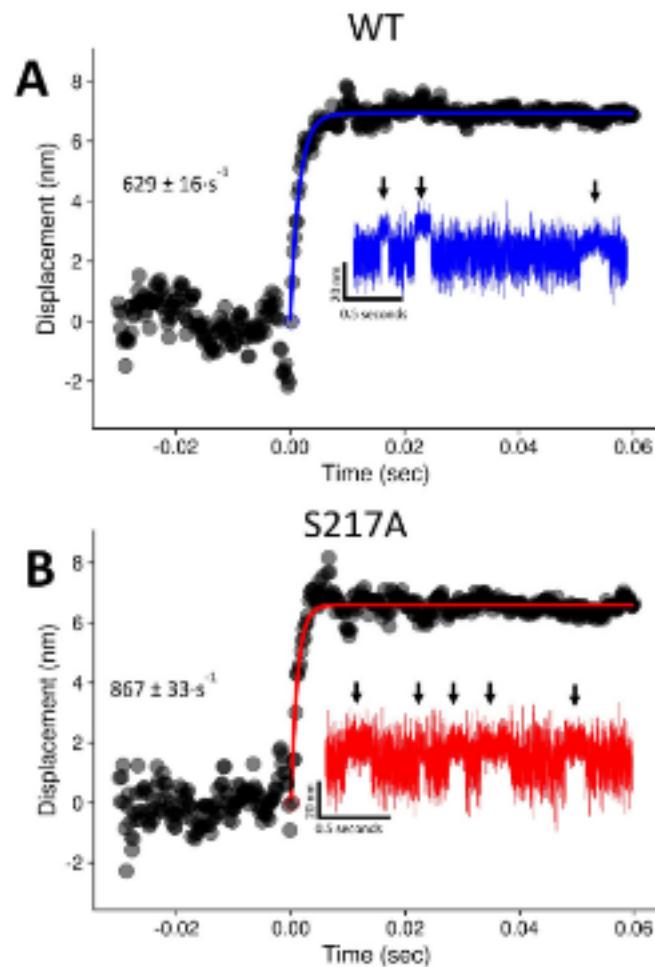


Figure 4.8. Ensemble averages created for Gunther 2020. Blue is WT myosin. Red is the S217A mutation. All data collected at pH 7.0 and 0mM-P_i. Raw traces are inset and ensembles are fit a with single exponential.

The major conclusion from the ensembles averages in Gunther et al. (2020) was that the mutation did not have a slowed transition from unbound-to-bound compared

to the WT (629 vs 867 s^{-1}) at 0mM-P_i concentration. If anything the mutation was slightly faster, which in hindsight I know believe this is due to the fact that the exponential was fit to the intial transition plus some additional data after the completion of the powerstroke. S217A was shown to have a $2\times$ fold faster ADP-release rate in Gunther et al. (2020) from solution kinetic studies, which coincidentally explains the faster attachments times we saw in this paper for the mutation. This faster ADP-release rate would also manifest in the hitch occuring sooner as compard to the WT. In the case of these forwards enemle averages, our exponential rate most likley reflect the rate of the initial transition plus some of the rate at which the hitch occurs. The S217A would have a higher percentage of events that would have been able to complete their hitch compared to the WT within the first 60ms . I think this is reflected in the slightly higher rate of the fit, but this would have no bearings/effects in regards to the validity of the analysis in answering the primary question.

4.2.5 Testing the Ensemble Averager

As described and shown in previous sections, the `{lasertrapr}` app has an analyzer for identifying events in single molecule laser trapping data. This analyzer determines the *exact* data point where an actomyosin binding event begins and the *exact* data point where the binding event terminates. With this information we can “temporally synchronize” the events and average them together to create ensembles that represent the average response of the binding interactions. A simple example of the process of the ensemble alignment procedure can simply explained with two theoretical events. Say we have two events that were both exactly 200ms , one event takes place from datapoints 5000 - 5999 and the other from datapoints $20,000$ - $20,999$. These datapoint indices would be returned by the analyzer as the respective beginning and ends of the events. They are both 1000 data points long making them 200ms long (assuming a sampling frequency of 5000 Hz). To average these events together,

we need to “temporally synchronize” the events. In the laser trap, the collected data we obtain is a time-series (the position of the bead is recorded over time) and the events occur at differing points in time as in the example (datapoint 5000 vs datapoint 20,000). Knowing the datapoint indices of where each event’s respective start occurs in time, datapoint 5000 for event 1 and datapoint 20,000 for event 2, the events can be subset out of the original raw trace, re-scaled, and placed on similar *relative* time scale. Meaning the start of event 1, datapoint 5000, becomes datapoint 0 and the last datapoint of event 1, datapoint 5999, becomes datapoint 999. We still have a 200ms long event which contain the same y-axis position values, just the x-axis values changed relative to the start of the event. This same procedure is performed with the second event. Datapoint 20,0000 becomes datapoint 0. Datapoint 20,999 becomes datapoint 999. Now we have 2 events with the same relative time scale. Now we can average the y-value associated with time 0 from event 1 with the y-value associated with time 0 from event 2 together, etc. This will create a new event that represents the average of the two. This can then be performed with how every many events are occurring in the datasets for each condition.

What makes this a little trickier than the example is that in reality every event is **not** the same length of time. The example provided prior walked through averaging two events of 1000 datapoints together, but what happens if one event was 1 second long and other was 200ms? This is actually the case when performing the analysis. During the alignment procedure the events need to be both 1) temporally synchronized starting at time 0, and 2) extended to the same length. Now in the case of 2 events that are 200ms and 1 second long, the 200ms event will be extended by 800ms. This can be accomplished by taking the average of the last few milliseconds of data within the event (datapoints 900-999) and obtain the average position value which can then be used to “extend” the event to 1 second long. Extending an event is adding/repeating data to the event to make it longer in time. So if the average y-value

of the last 2 milliseconds of the forward ensemble of event 2 is 6.7nm. Then that value (6.7) is added 4000 times to the event 1 data to result in having 2 events that are both 1 second long (assuming 5000hz, 1 second is 5000 datapoint, 200ms is 1000 datapoints. $5000 - 1000 = 4000$ datapoints to pad/concatenate/add/extend/however you want to phrase it). The events can then be averaged together as described previously.

{lasertrapr} has the functionality to create ensemble averages which includes auto-generated plots and a choice of a single, double, or no exponential fits to the data. In order to test the accuracy of the app in the construction of ensemble averages two distinct datasets were simulated. In one data set, every event had exactly a 4nm step size with an accompanying 2nm hitch while the other has a 7nm step with a 3nm hitch (plus brownian noise). No displacement distribution were simulated, every event was a positive 4 or 7nm. Both the ATP and ADP binding rates were 10 set at 10s^{-1} for the 4+2 group and 5s^{-1} for the 7+3 simulations. Making the ensembles in the app is a three part process. The first step is creating the ensembles, averaging/fitting, and plotting. The app allows some user selection in creating the ensembles in regards to selecting a time period to extend the events forwards/backwards.

I have found extending the backwards ensembles to be a little tricky since the single molecule analysis classifies the transition into an even as part of the event. For example, since each event is a summation of myosin's powerstroke, say its a 6nm step and 2nm hitch (plus brownian motion) then the total displacement would be 8nm (d_{total}). This can be broken down into distinct displacement substeps. d_1 which would represent the displacement caused by the intial powerstroke, would be 6nm and d_2 the additional displacement accompanying the hitch would be 2nm. If you took the average position of the first 1ms, you would be averaging the transition from unbound to bound, and extending backwards in time with a value that would be smaller than d_1 (6nm). This would cause an accidental inflation of the size of the hitch in the backwards ensemble, or if you calculated the hitch size simply by

differencing the forwards/backwards ensembles. To properly extend the backwards ensemble you need to average the true post-powerstroke, pre-hitch displacement. The d_1 state. In the app this is accomplished by “skipping” into the event before averaging the position. The idea is to be able to skip through the unbound/bound transition into the d_1 state so your average value for backwards extension is a true representation of the d_1 state to increase accuracy of the ensembles.

Extending the forwards ensembles is more straightforward since the single molecule analyzer does not tend to classify the transition back to baseline as part of the event so the last datapoint that signifies the end of the event in the analysis tends to be at peak displacement, so the app can just average the few milliseconds prior to this datapoint to represent the d_2 post-hitch displacement and extend forward. The one consideration to keep in mind when extending forward is to be diligent of the ATP concentration the experiments were performed at and/or the ATP affinity of the motor used. At infinitely high ATP concentrations with a motor with an infinitely high ATP affinity creating an ensemble average perhaps may not be possible. Assuming that the hitch accompanies ADP-release, the rigor state would then be infinitely short/undetectable because the motor would spend too little of time at the post-hitch d_2 final displacement segment. This would result in (I think) being unable to properly extend forward the events because there would no d_2 datapoint to average/extend. The extension would be from the d_1 displacement which would create a forward ensemble that would appear to have no hitch solely because there would be no data to properly create the ensembles with. The app allows the user to select the time period (in milliseconds) to extend forward. You need to be a little more aggressive with selecting a short time period when dealing with assays at higher ATP. I think lower ATP conditions are a little more forgiving since there should be a longer rigor state where the motor is spending time/collecting datapoints at the final post-hitch d_2 state. I believe you have a high probability of

properly averaging the correct position to extend forward with in these lower ATP conditions.

Since there are parameters that the user can fiddle around with, along with different equation to fit the ensembles with, it seemed better to be able to do those things separately in the app so you can create prepare the ensembles by specifying alignment procedure in one step and then perform the actual averaging/fitting in another. This allows the user to change options independent of each other which is nice because the former part generally takes longer to compute, and you don't have to spend as much time sitting around waiting for your computer to prep the ensembles.

The resulting ensemble averages of the simulated data are shown in Figure XX. The analysis was able to estimate a 3.7nm step and 2.3nm hitch for the 4+2 conditions, and a 6.9nm step and 3.2nm hitch was estimated for the 7+3 conditions. Note that the estimations of the d_1 and d_2 values are coming from the double exponential fits to the forwards ensmebles. The unbound/bound transition is simulated as occurring instantaneously in the app's simulator so the initial transition rate is quite fast ($>500/\text{s}$) which reflects what would be seen in real data which where that rate is also usually around $500/\text{s}$ and is dependent on the corner frequency of the instrument reflecting the average rate of a bead being damped by viscous drag forces.

Interestingly, the rates of k_1 which are associated with the rate of ADP release closely match the rates that were input into the simulations. The double exponential fits to the forwards ensembles estimate a rate of $13/\text{s}$ where the value of $10/\text{s}$ was input as the rate of ADP release of the 4+2 conditions, whereas the fits estimated an ADP release rate (k_1) as $6/\text{s}$ for the 7+3 conditions and $5/\text{s}$ was the rate input.

The backwards ensembles are always fit with a single exponential which also estimates the d_1 and d_2 values from the fit parameters along with the rate of k_2 which is generally attributed to the ATP binding constant. For these simulations the ATP binding constants were set at $10/\text{s}$ and $5/\text{s}$ for the 4+2 and 7+3 conditions respec-

tively and the single exponential fits to the backwards ensembles estimated these rates to be 7.4/s and 5.0/s.

Additionally, non-linear fits are not necessary to estimate the size of the hitch, you can opt for a simpler method and compute the difference between the end of the forwards ensembles and the beginning of the backwards ensembles to estimate the size of the hitch. A rough estimate of this can be made “by eye” by looking at Figure 4.8. First examine the forward ensembles final displacement on the graph and it appears to be around 6nm as shown by the blue data. You can then see that the backwards ensembles start at a value just around 4nm (maybe slightly less). So the final displacement of 6nm minus 4nm equals our 2nm hitch which would be the same computed by the fits, albeit without the accompanying rate constants.

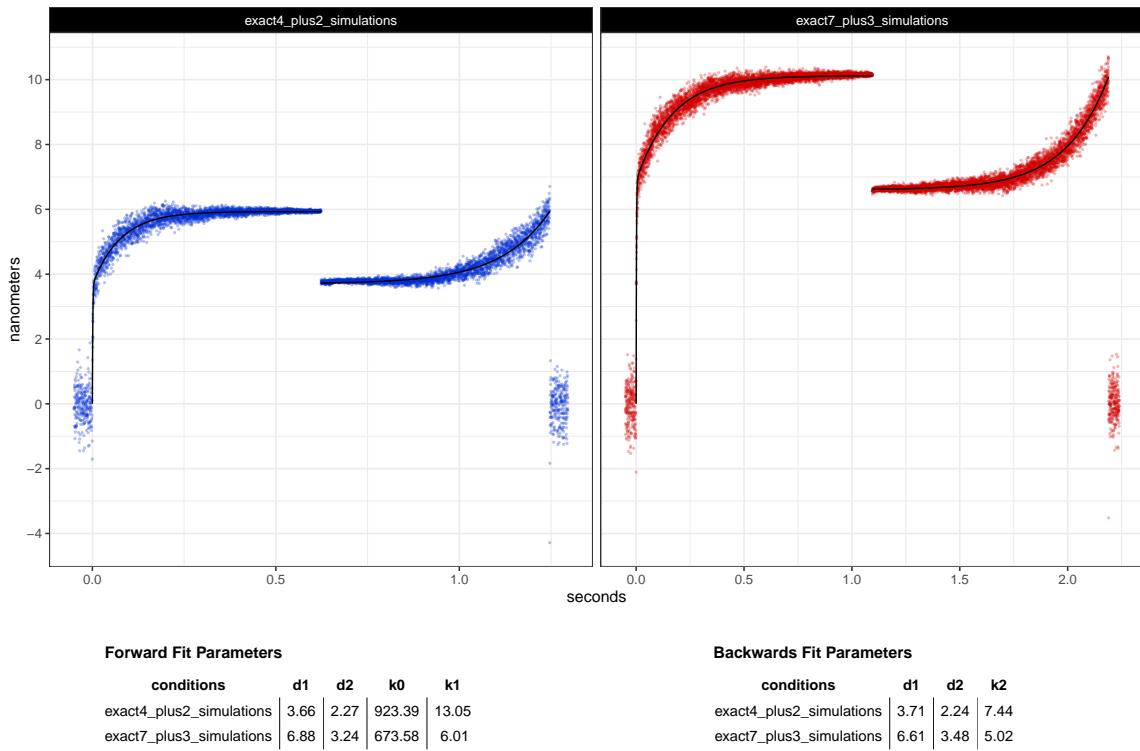


Figure 4.9. Ensembles averages of the simulated datasets for validation. The tables show values from the fits that closely correspond to the true values.

4.2.6 Mini-Ensemble Analyzer

An additional feature of the `{lasertrapr}` app is the ability to analyze mini-ensemble laser trap data. The mini-ensemble assay is the exact same assay as described for these studies performed for this dissertation (Ch. 5/the Cytoskeleton paper) except with more myosin on the surface. In the mini-ensemble laser trap assay multiple myosin heads can interact/bind to a single actin filament causing rapid displacement, or “runs”. This assay displays the ability of a team of molecule motors to create larger ensemble forces when allowed to work together. The mini-ensemble analyzer in the app was a recreation of prior published work from the lab (Longyear, Walcott, and Debold (2017)). Briefly, raw data is low-pass filtered using a running mean with a window width that can be variably set by the user (typically ~10 ms, 50 data points at the sampling rate of 5kHz). Events are then identified using two criteria: 1) a user defined displacement threshold is set to the running mean to signal the start or end of an event, and 2) the event meets/exceeds a minimum defined attachment time. Attachment duration can then be calculated as the time between the start and end displacement thresholds. The time between events can be calculated as the time between the end of an event and the start of the subsequent event. Peak forces can be estimated by identifying the maximal displacement of each record and converting the displacement into forces, by multiplying the peak displacement by the combined trap stiffnesses.

As a matter of fact, the mini-ensemble analyzer was the first analyzer I ever wrote for the lab. This analyzer was used in Woodward et al. (2020) to compare the effects of an abiotic triposphate compound that was altering myosin’s behavior. The data analysis helped provide invaluable insight into understanding the mechanisms of the decreased force producing capabilities of myosin when it used the ATP alternative compound as substrate.

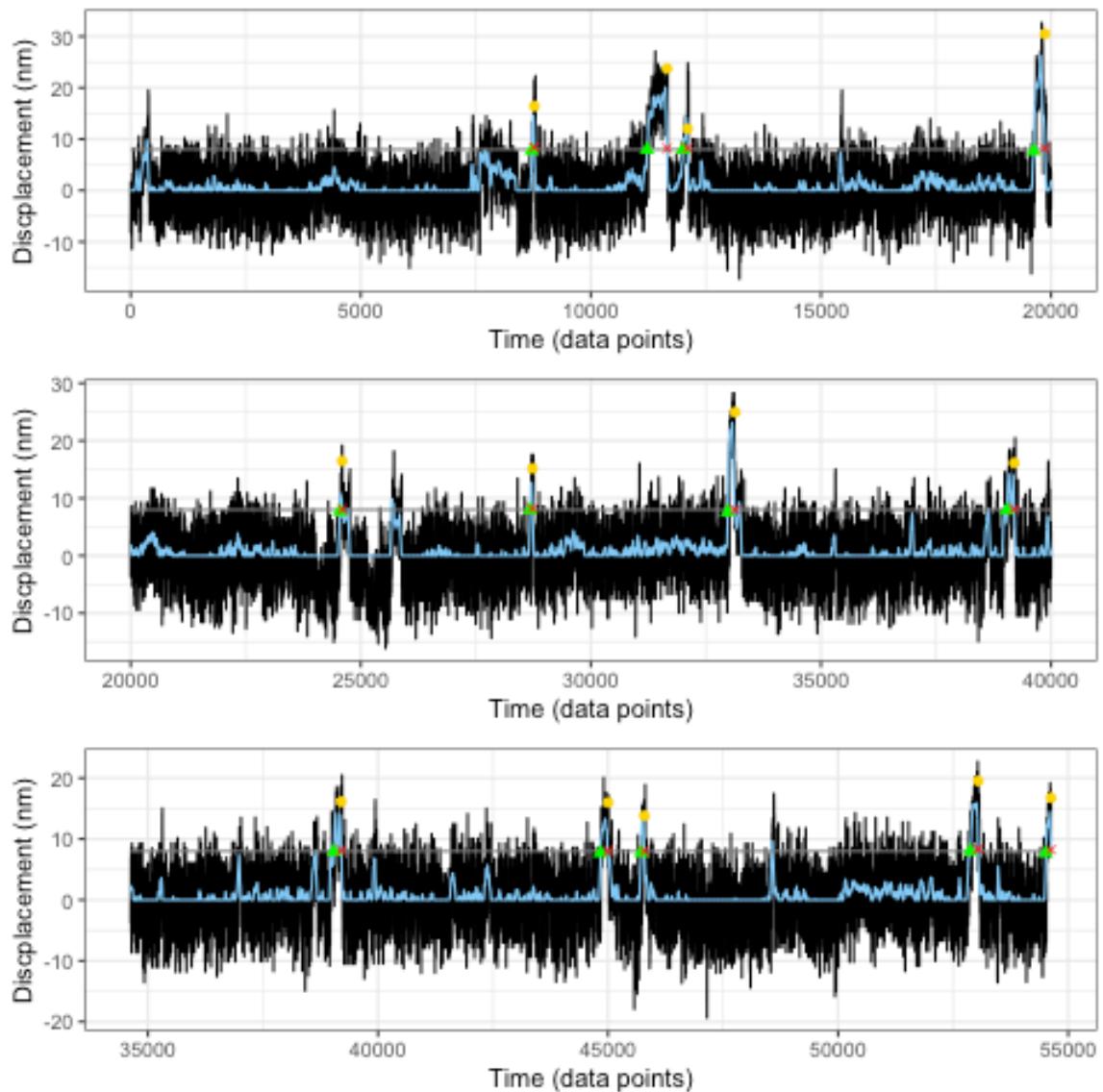


Figure 4.10. An early output plot from the mini-ensemble analyzer. The plots are now scrollable/interactive, but this serves as a fun reminder of where the app began in its infancy years prior to its creation.

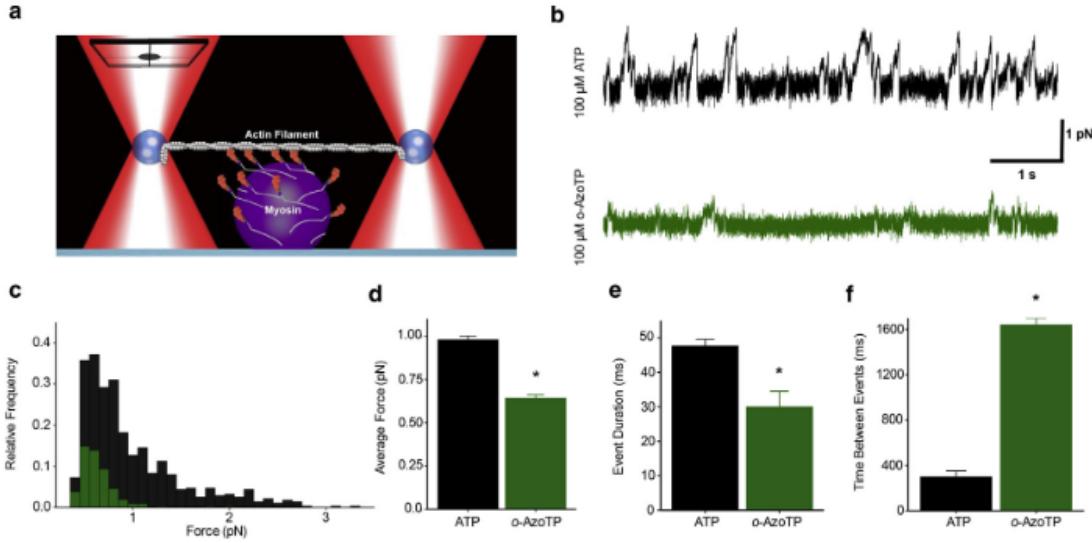


Figure 4.11. Image from Woodward et al. (2020) showcasing the mini-ensemble analyzer that is now a core feature within the app.

4.2.7 Single Molecule Event Identification (step-by-step)

This is a “step-by-step” walk through of the Hidden-Markov/Changepoint Analysis we use to analyze our single molecule laser trap data and includes everything on the journey from raw data to analyzed trace and everything on the way...buckle up. This section is a lot cooler if you are reading online and includes the R code to reproduce this by hand. You can download the data [here](#)

4.2.7.1 Raw data

Here is a raw data trace. This is unprocessed data as-is from the trap computer. The data is relative position of the bead in mV over time:

The data record is 89.1576 seconds long and has an average position of 6.2968527 mV.

4.2.7.2 Processed Data

The first step of the analysis is removing the “baseline” by centering the mean around 0. This can either be done by simply subtracting the baseline mean from

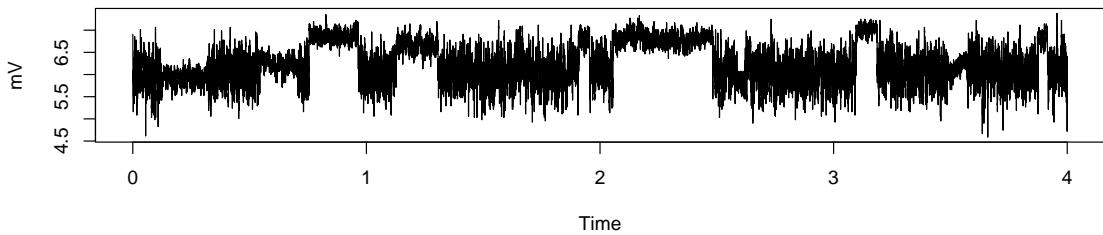


Figure 4.12. Raw trap data.

every data point or by performing a piecewise linear detrend on the whole record. The latter accomplishes two things: **1)** Centers mean around 0 and **2)** removes any drift (i.e. wander correction). Additionally, in the `{lasertrapr}` app you can find the average baseline position by using a mean variance transformation of the data to select the baseline population or by selecting a quiescent period of the data where no binding events occur to calculate mean baseline position. Here we will detrend the data and convert from mV to nm using a “Step Calibration”. The step calibration is performed by moving the microscope stage a known distance, say 200 nanometers, and measuring the resulting change in the mV signal. We then can estimate the number of nanometers per mV.

The mV-to-nm conversion calibrated around the time this data trace collecting we are analyzing know was 30 nm/mV. We can convert our raw data from mV to nm and detrend the data (or visa versa).

4.2.7.3 Running Mean & Variance

Running Mean & Variance

The next step is to transform the data for the HM-Model by calculating both the running mean and variance of the data trace. This analysis uses a 150 datapoint

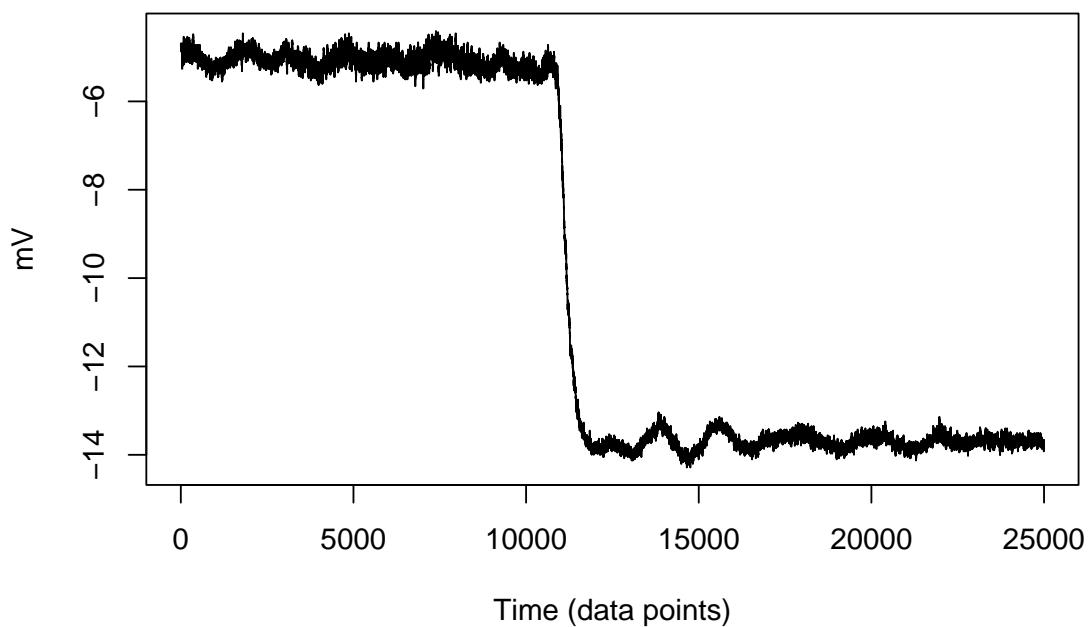


Figure 4.13. Example of a step calibration. Stage was moved 200nm

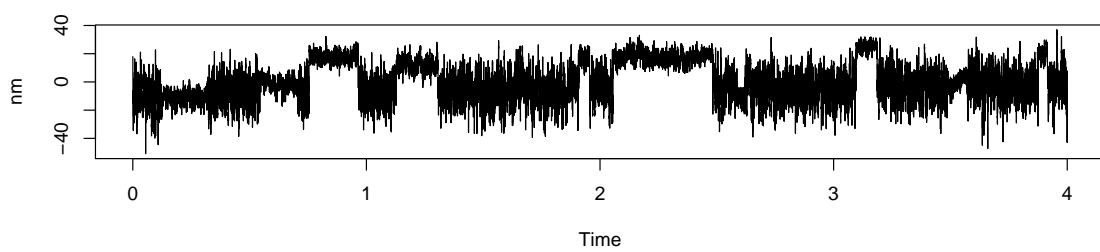
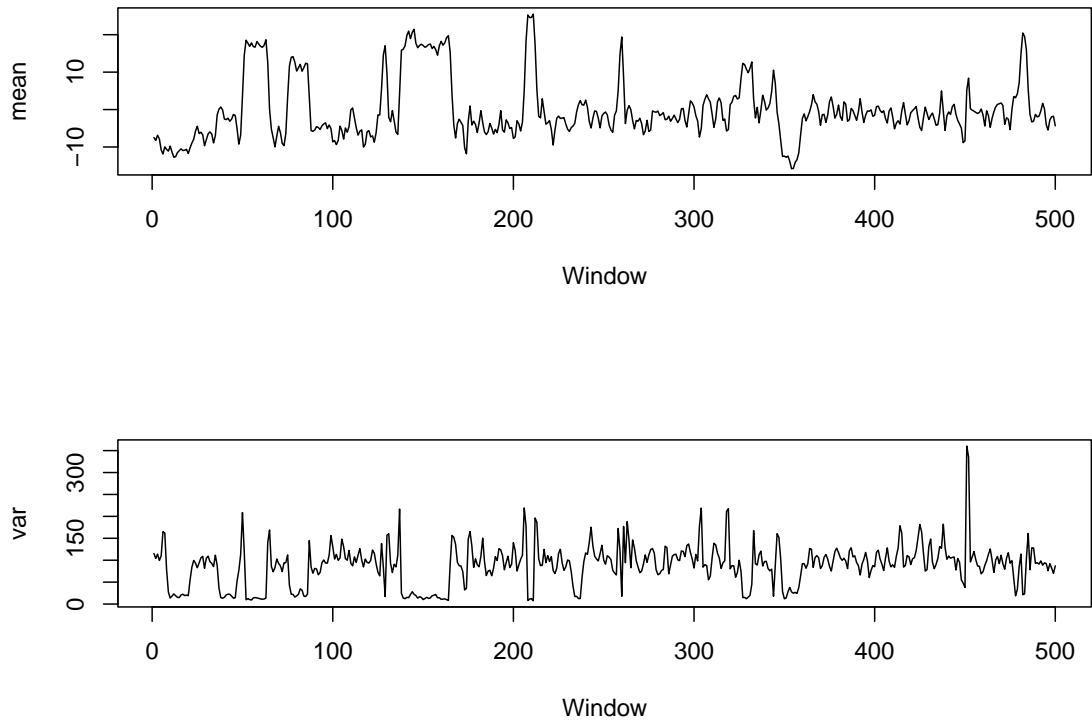


Figure 4.14. Example of processed trap data

window that advances by sliding 75 data points each time (half the window width). This is done to decrease the correlation between neighboring windows:



Additionally, we can plot these datasets against each other to see the mean vs. variance for each window:

4.2.7.4 Hidden-Markov Model

The data is now ready to have events identified with a Hidden-Markov Model. We first need to initialize the model with guesses of the initial state probabilities and the transition probabilities for our 2-state model. State 1 is when myosin is unattached from actin and state 2 is attached. We need to guess 6 different numbers.

- 1) Probability of initially starting in state 1
- 2) Probability of initially starting in state 2
- 3) Transition probability from going from state 1 to state 1
- 4) Transition probability from going from state 1 to state 2

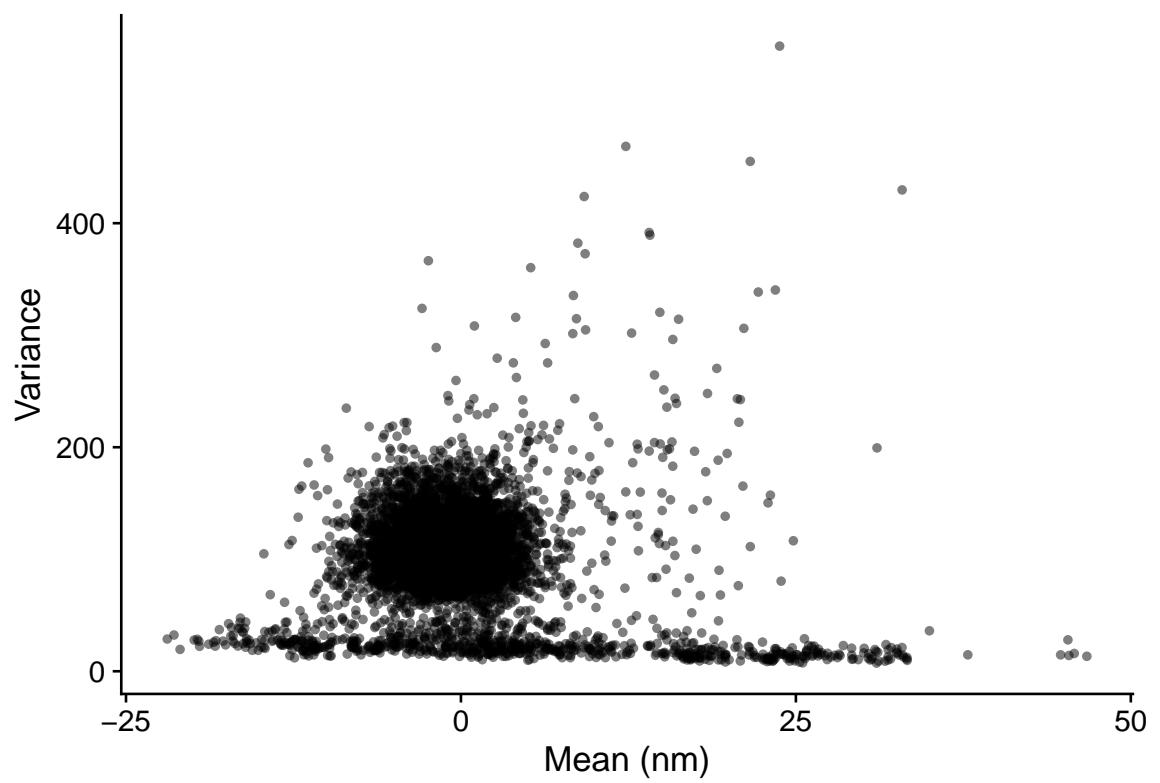


Figure 4.15. Mean/Variance Plot

- 5) Transition probability from going from state 2 to state 1
- 6) Transition probability from going from state 2 to state 2

I prep the data so it usually always start with baseline (i.e. state 1) or will trim it so the trace does though 99.9% this just occurs so we will give guess that 98% probability of starting in State 1. Probability of starting in state 2 is then $1 - [\text{Prob. S2}] = -.02$. I then assume that these are both stable states and that there is a high probability of transitioning from state 1 to state 1 or state 2 to state 2. By the same logic above the transition probabilities are guess and our 6 probabilities above are:

```
## [1] 0.98 0.02 0.98 0.02 0.02 0.98
```

We will also have to make guesses of the statistical characteristics of the 4 underlying Gaussian distributions (2 states for each the running mean and running variance). To do this we will estimate the mean and standard deviation of each of the Gaussian. The 8 numbers that follow are:

- 1) Guess for the mean of the variance for State 1. -calculated by taking the mean of the running variance
- 2) Guess for the sd of variance for State 1 -calculated by the taking the sd of the running variance
- 3) Guess for the mean of mean for State 1 -hard coding to 0 because we centered baseline around 0 when we processed the data.
- 4) Guess for the sd of mean for State 1 -calculated by taking the sd of running mean
- 5) Guess for the mean of variance for State 2 -estimated as half the value as state 1 (signal-to-noise 2:1)

- 6) Guess for the sd of variance for State 2 -#2/2 because of stiffer system when myosin attached
- 7) Guess for the mean of mean for State 2 -hard coded at 5nm (estimated size of the powerstroke)
- 8) Guess for the sd of mean for State 2 -coded as twice #4 since there will be positive and negative displacements

```
## [1] 101.69 45.56 0.00 6.47 50.85 22.78 5.00 6.47
```

Once we estimate the starting point the model can be fit. The HM-Model will optimize all the parameters we just defined using the Expectation-Maximization (EM) Algorithm. The resulting model summary (Re1. = “Response 1” and is the variance signal while Re2. = “Response 2” and is the mean signal):

```
## converged at iteration 12 with logLik: -47137.21

## Initial state probabilities model
## pr1 pr2
##   1   0
##
## Transition matrix
##           toS1  toS2
## fromS1  0.974 0.026
## fromS2  0.180 0.820
##
## Response parameters
## Resp 1 : gaussian
## Resp 2 : gaussian
##      Re1.(Intercept) Re1.sd Re2.(Intercept) Re2.sd
```

```

## St1      113.302 36.224      -0.758  3.950
## St2      22.375  8.612       5.185 13.816

```

Starting from the first line in the summary we can see that the model gives a 100% probability that this record starts in baseline (state 1). The transition matrix is straightforward to read and these probabilities could also be used as estimates of on/off rates although currently we have not been using these. Lastly, the “response parameters” are the optimized characteristics of the Gaussian distributions describing each of the state 1 and state 2 normal distribution for both the mean and variance signal. The results are that the variance distribution for state 1 has a mean of 113 (SD of 36) and state 2 has a variance of 22 (SD of 8). This gives a signal to noise of a little more than 5:1 in the variance signal. Furthermore, the baseline mean has a mean of -0.7 (SD of 4) and state 2 mean is 5.1 (SD of 13).

Now knowing these parameters we can extract the most probable state sequence through the trace via the Viterbi Alogorithm. Here is a table of the posterior states (columns are state = viterbi state, S1 & S2 are the delta probabilities of being in each state):

state	S1	S2
1	1	0
1	1	0
1	1	0
1	1	0
1	1	0
1	1	0

We can now construct another Mean/Variance plot using the running window transformation and this time color code the windows by State:

At this point I think we have completed the hardest part of laser trap analysis - **picking events**. Now we need to find the the peak displacements (and subsequently

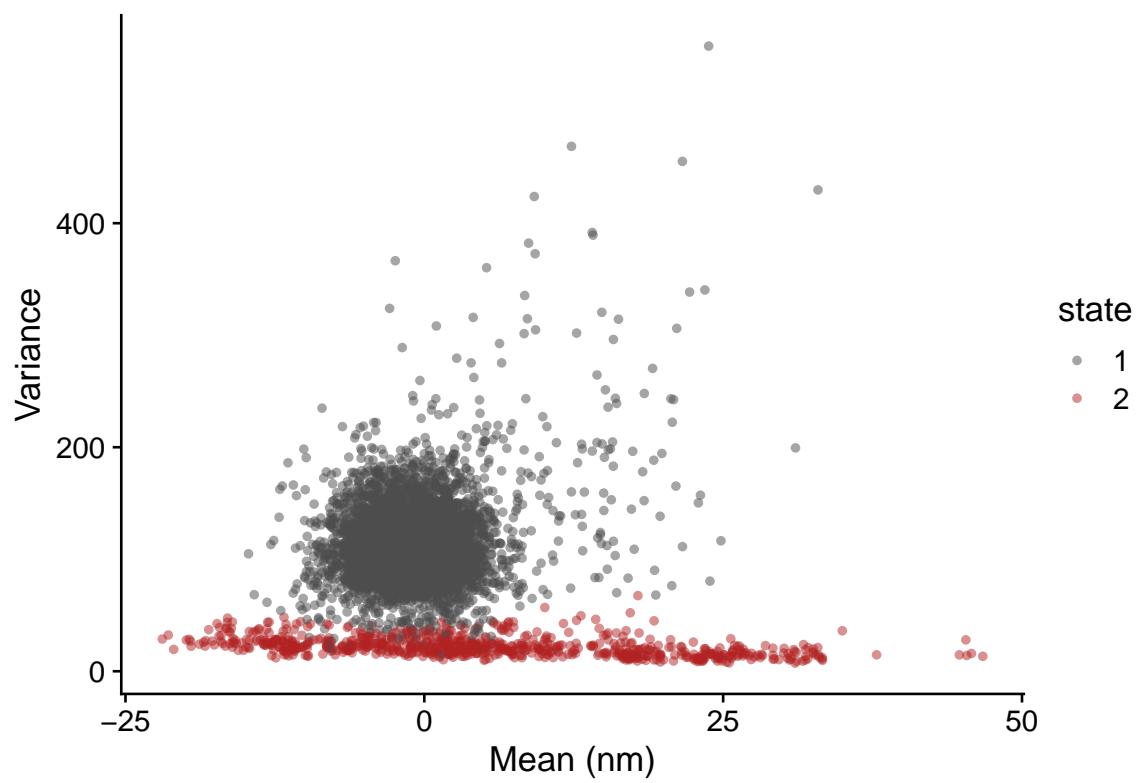


Figure 4.16. Mean/Variance Plot colored by State

forces), time off, and time ons for each event. I call this ‘measuring events’ (because in reality I feel like these could be simple measurements you could make with a ruler).

4.2.7.5 Measure Events

The HM-Model state sequence decoding assigns every running window a 1 or a 2 describing the state that the window most likely belongs to. So all we are given is a long list of a 1-2 indicator that is the same length as our running mean/variances.

We need to calculate the number of 1’s and 2’s that occur in a row and then calculate the cumulative sum of these ‘in-a-row’ counts to get the indices of when the events start/stop in running window time. Here is the number of 1’s and 2’s that occur in a row.

lengths	values
8	1
12	2
16	1
10	2
5	1
12	2

The table can be read that the raw trace starts with 8 windows of state 1 baseline noise followed by 12 windows of a state 2 event etc.

On times in milliseconds can be estimated by taking those state 2 ‘in-a-row’ lengths and converting them to 5kHz time and then to ms. The conversion from window time to 5kHz can be calculated by dividing the length of the raw/processed data by the length of the resulting running mean/var calculations:

```
## The conversion between raw data and running windows is 75
```

This also works out to be the advancing window sliding distance in the running mean/var calculations. Estimating on-times is then straightforward. Multiply state

2 ‘in-a-row’ length by the conversion and then divide by sampling frequency (5000) to get into seconds and multiplied by 1000 for ms.

The same idea can be applied to the state 1 baseline to get the off times. In the process the first and last off-times are excluded because in reality we do not know when the last event before our first recorded event actually occurred. Additionally, the last baseline/state 1 ends because we stop recording so that is also not a true measure. Here is the table of on/off times:

n_event	hmm_state	num_windows	length_5kHz	time_on_ms	time_off_ms
1	2	12	900.27869	180.05574	NA
2	2	10	750.23225	150.04645	240.07432
3	2	12	900.27869	180.05574	75.02322
4	2	11	825.25547	165.05109	180.05574
5	2	1	75.02322	15.00464	630.19509
6	2	27	2025.62706	405.12541	120.03716

In this table the *time_off_ms* column refers to the off time that occurred prior to the event.

Moving along and to make this information more helpful in being able to really ID where the events are stopping and starting we can take the cumulative sum of these ‘in-a-row’ lengths that will give us the running window indices of the start/stop of the events. This will help us chunk out the events to measure step sizes. After calculating the cumulative sum of the ‘in-a-row’ lengths we can make a new table with 2 columns. One column *state_1_end* indicating the window which is the last window in a series of baseline and a *state_2_end* column that is the last window of an event series.

state_1_end	state_2_end
8	20
36	46
51	63
75	86
128	129
137	164

The result is the running window indices of the window before the event starts and the window that ends the event. By adding one (+1) to the *state_1_end* value we get the index of the start of each event.

```
## So the first event is between indices 9 and 20
```

We can estimate the step size from the running windowed data. To get the step size we find the position of the running window with the greatest absolute value and take its real value. Finding the index of the window with the greatest absolute value let's us also find the peak of negative events more accurately than just taking the max right off the bat.

Now this would give us step size estimates since we already processed the data to be centered around 0. However, our baseline signal does not always return to 0 after an event. To get a more precise estimate of our step sizes by the same logic we can calculate the average of the baseline noise prior to each event and then subtract the step size estimate from the baseline prior to the event.

This results in a table giving the mean of the baseline before each event (*avg_s1*), the estimated step size (*avg_s2*), and the differences between the 2 (*diff*) representing the final step size the program reports for each event.

avg_s1	avg_s2	diff
-9.2145628	-12.734709	-3.520146
-6.9028608	-3.127884	3.774977
-0.6201406	18.654526	19.274667
-4.1346721	14.146184	18.280856
-4.5201374	17.046019	21.566157
-0.9125085	21.452987	22.365496

4.2.7.6 Direction Correction

Admittedly, sometimes the actin filament is oriented in the wrong direction. To compensate for backwards filaments the program has a so-called ‘direction-correction’ and will auto-magically flip the raw trace if the filament was oriented the wrong way. What happens here is after analyzing step sizes if the program identified more negative events than positive events it assumes backward-filament orientation in the trap and flips the trace horizontally over the x-axis by multiplying every value by negative one (-1). Now if the records were flipped so were the calculated step sizes and these new values reported along with the force measures that are calculated by the step size multiplied by the user-inputted nm -> pN conversion. These values are added to the on/off time table previously calculated.

4.2.7.7 Changepoint Analysis

Since we identified the events we can also perform ensemble averaging with a little more work. Unfortunately, preparing the data for the HM-Model by transforming into running mean/variances decreased our time resolution as we lost a lot of information. Our original number of data points collected is 445788, whereas the number of data points (windows) in the running mean/var is 5942.

The resulting start/end of each events are really just estimates. To get better start of event estimates the program uses changepoint analysis on the transition periods

into and out of each event to better estimate the start of each event in the original sampling frequency.

To obtain the transition periods from raw data in 5kHz time the running window indices from the state_1_ends and state_2_ends are converted to 5kHz time by the conversion ratio of ~ 75 we previously calculated as the ratio between the number of datapoints in the raw data to the data points in the running mean/var calculations.

More specifically, the program finds the running window index of ~ 1.5 windows back into the baseline from the *state_1_end* window and the index of the first state 2 window and converts back into 5kHz time to supply a slightly larger transition window to analyze for the ‘true’ event start. Doing this for every event we obtain a new table with 5kHz time indices that *should* contain the transition into every event:

The changepoint analysis is actually performed on a new running variance of the processed data with a variable length window width. Here I use 5 datapoints (1ms) because the trace has exceptional signal-to-noise. This running transformation advances one point at a time. The resulting running variance transform looks like this:

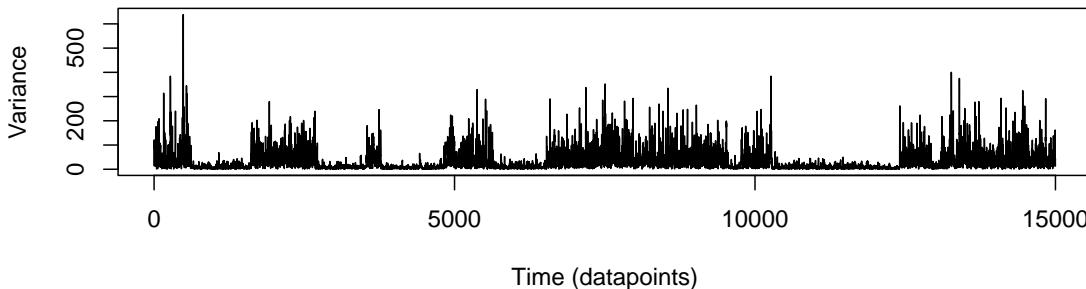


Figure 4.17. 1ms running variance transformation

We can then chunk out the the transition periods with our indices and apply changepoint analysis to each transition period that looked for the change in mean

of the variance signal for every event and plot the results. The changepoint analysis looks for a change in mean of the variance signal. The analysis only looks for a single changepoint. *Note, this is an early version of the changepoint analysis that {lasertrapr} used. See the source code of the app on GitHub for up to date version.*

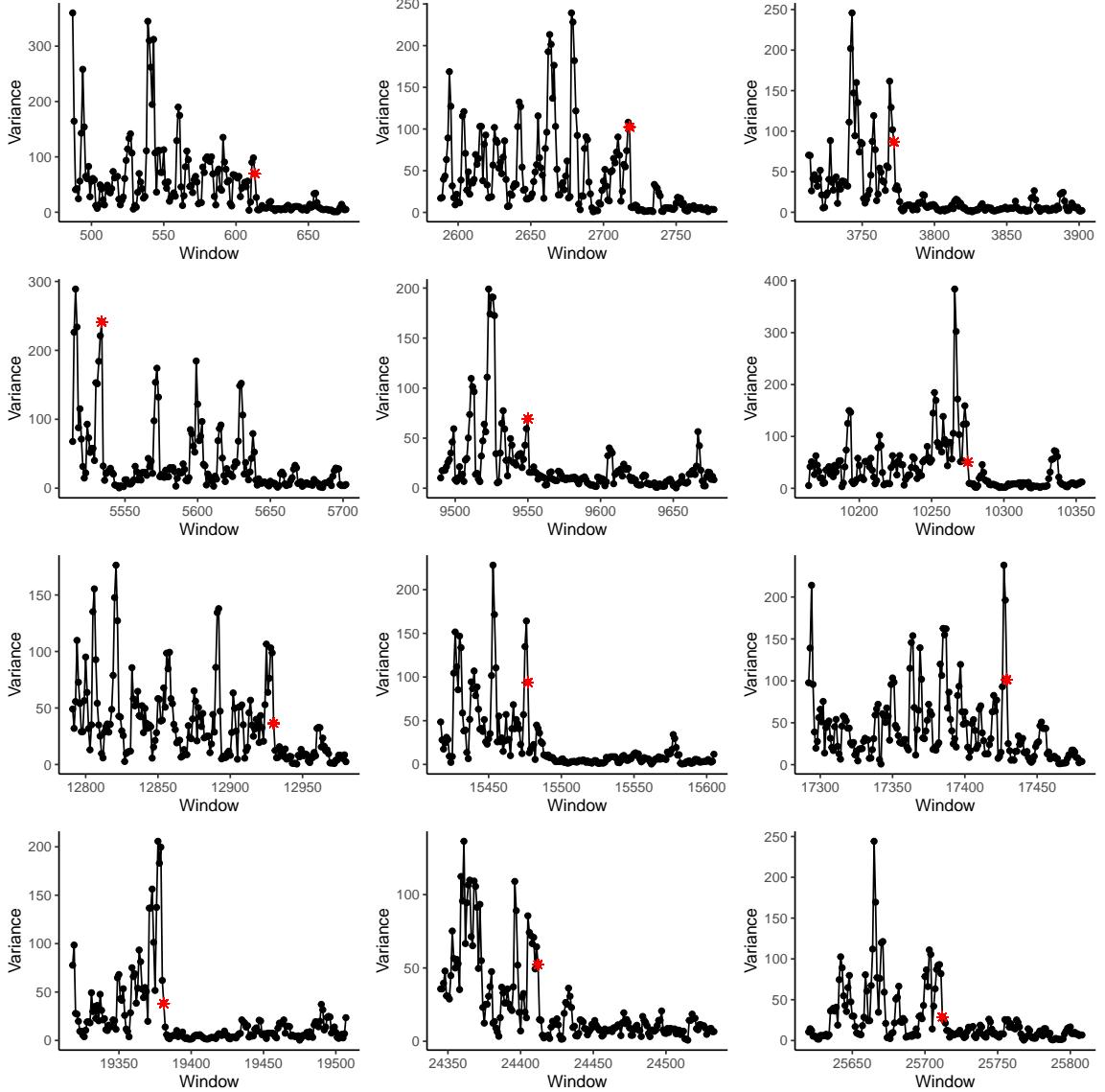


Figure 4.18. Changepoint identified start of event in running variance

We can also plot the corresponding point in the raw/processed data that is ultimately used in the ensemble average:

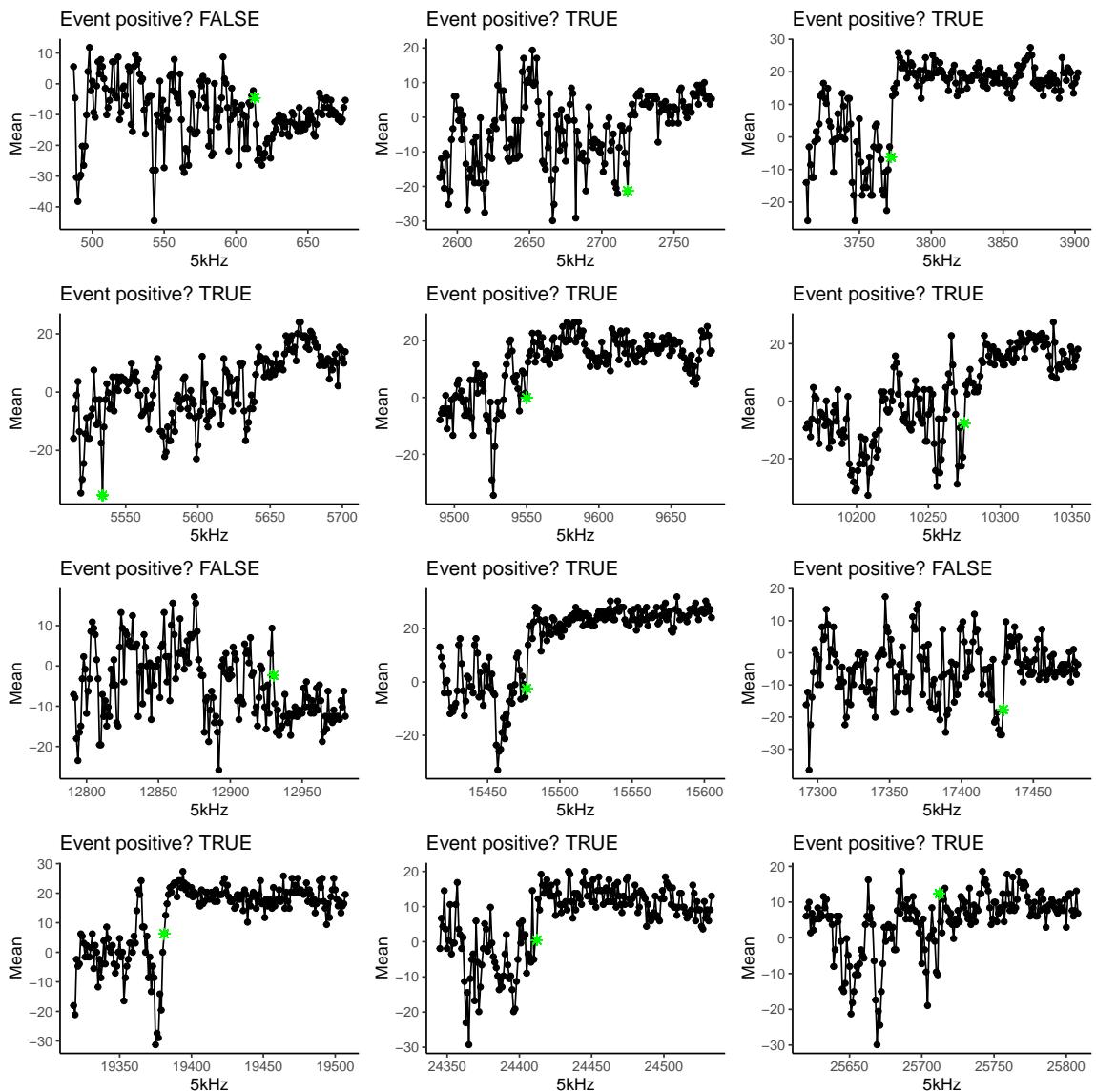


Figure 4.19. Changepoint identified start of event in the original data set

This same approach can be applied to the backside. Once backside change points are identified than more precise measurements of time on, time off, displacements, and forces can be estimated. For now, we can go ahead and make one final plot showing the complete analysis overlayed on the raw/processed data. The pink shades indicates the start/end of the event. The vertical dashed lines are placed at the peak displacement of each event and labeled with the step size and event duration. No analysis is perfect, some shorter events are missed. These are usually running windows which have a high variance due to the window overlapping baseline and an event.

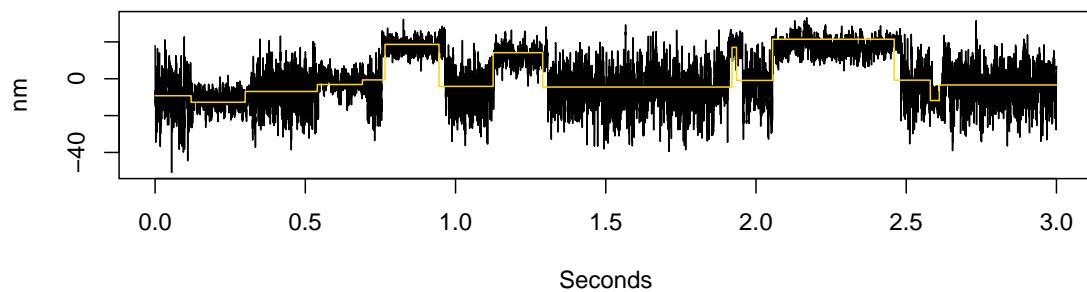


Figure 4.20. Results of analysis overlayed on data trace

Moving along and to make this information more helpful in being able to really ID where the events are stopping and starting we can take the cumulative sum of these ‘in-a-row’ lengths that will give us the running window indices of the start/stop of the events. This will help us chunk out the events to measure step sizes. After calculating the cumulative sum of the ‘in-a-row’ lengths we can make a new table with 2 columns. One column *state_1_end* indicating the window which is the last window in a series of baseline and a *state_2_end* column that is the last window of an event series.

state_1_end	state_2_end
8	20
36	46
51	63
75	86
128	129
137	164

The result is the running window indices of the window before the event starts and the window that ends the event. By adding one (+1) to the *state_1_end* value we get the index of the start of each event.

```
## So the first event is between indices 9 and 20
```

We can estimate the step size from the running windowed data. To get the step size we find the position of the running window with the greatest absolute value and take its real value. Finding the index of the window with the greatest absolute value let's us also find the peak of negative events more accurately than just taking the max right off the bat.

Now this would give us step size estimates since we already processed the data to be centered around 0. However, our baseline signal does not always return to 0 after an event. To get a more precise estimate of our step sizes by the same logic we can calculate the average of the baseline noise prior to each event and then subtract the step size estimate from the baseline prior to the event.

This results in a table giving the mean of the baseline before each event (*avg_s1*), the estimated step size (*avg_s2*), and the differences between the 2 (*diff*) representing the final step size the program reports for each event.

avg_s1	avg_s2	diff
-9.2145628	-12.734709	-3.520146
-6.9028608	-3.127884	3.774977
-0.6201406	18.654526	19.274667
-4.1346721	14.146184	18.280856
-4.5201374	17.046019	21.566157
-0.9125085	21.452987	22.365496

4.2.7.8 Direction Correction

Admittedly, sometimes the actin filament is oriented in the wrong direction. To compensate for backwards filaments the program has a so-called ‘direction-correction’ and will auto-magically flip the raw trace if the filament was oriented the wrong way. What happens here is after analyzing step sizes if the program identified more negative events than positive events it assumes backward-filament orientation in the trap and flips the trace horizontally over the x-axis by multiplying every value by negative one (-1). Now if the records were flipped so were the calculated step sizes and these new values reported along with the force measures that are calculated by the step size multiplied by the user-inputted nm -> pN conversion. These values are added to the on/off time table previously calculated.

4.2.7.9 Changepoint Analysis

Since we identified the events we can also perform ensemble averaging with a little more work. Unfortunately, preparing the data for the HM-Model by transforming into running mean/variances decreased our time resolution as we lost a lot of information. Our original number of data points collected is 445788, whereas the number of data points (windows) in the running mean/var is 5942.

The resulting start/end of each events are really just estimates. To get better start of event estimates the program uses changepoint analysis on the transition periods

into and out of each event to better estimate the start of each event in the original sampling frequency.

To obtain the transition periods from raw data in 5kHz time the running window indices from the state_1_ends and state_2_ends are converted to 5kHz time by the conversion ratio of ~ 75 we previously calculated as the ratio between the number of datapoints in the raw data to the data points in the running mean/var calculations.

More specifically, the program finds the running window index of ~ 1.5 windows back into the baseline from the *state_1_end* window and the index of the first state 2 window and converts back into 5kHz time to supply a slightly larger transition window to analyze for the ‘true’ event start. Doing this for every event we obtain a new table with 5kHz time indices that *should* contain the transition into every event:

The changepoint analysis is actually performed on a new running variance of the processed data with a variable length window width. Here I use 5 datapoints (1ms) because the trace has exceptional signal-to-noise. This running transformation advances one point at a time. The resulting running variance transform looks like this:

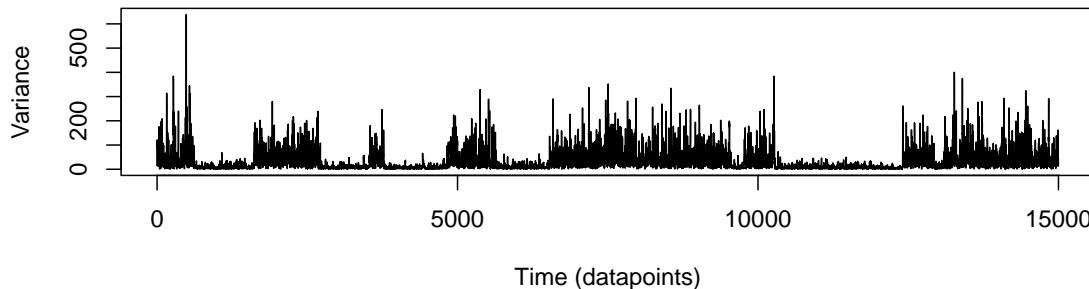


Figure 4.21. 1ms running variance transformation

We can then chunk out the the transition periods with our indices and apply changepoint analysis to each transition period that looked for the change in mean

of the variance signal for every event and plot the results. The changepoint analysis looks for a change in mean of the variance signal. The analysis only looks for a single changepoint. *Note, this is an early version of the changepoint analysis that {lasertrapr} used. See the source code of the app on GitHub for up to date version.*

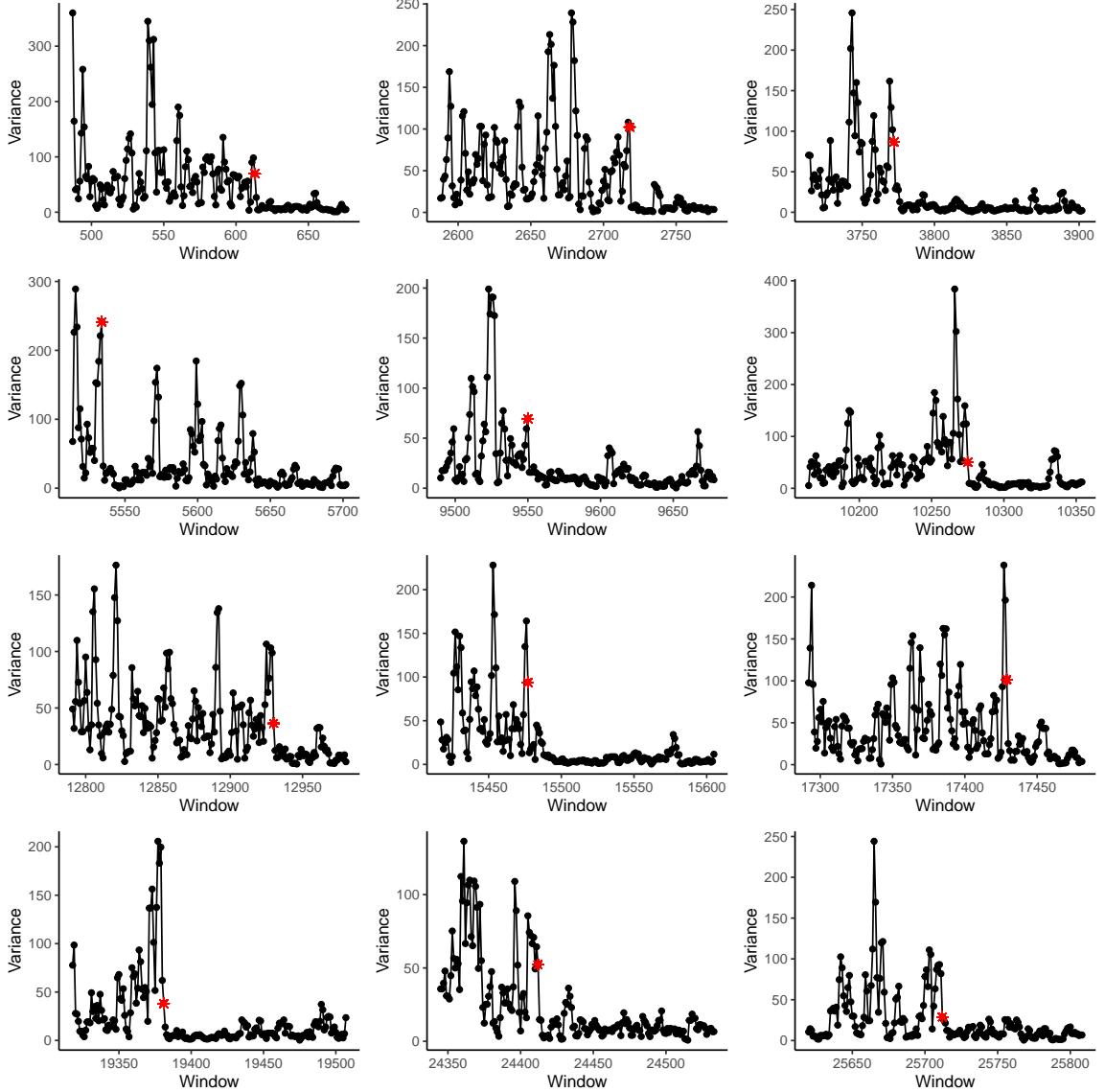


Figure 4.22. Changepoint identified start of event in running variance

We can also plot the corresponding point in the raw/processed data that is ultimately used in the ensemble average:

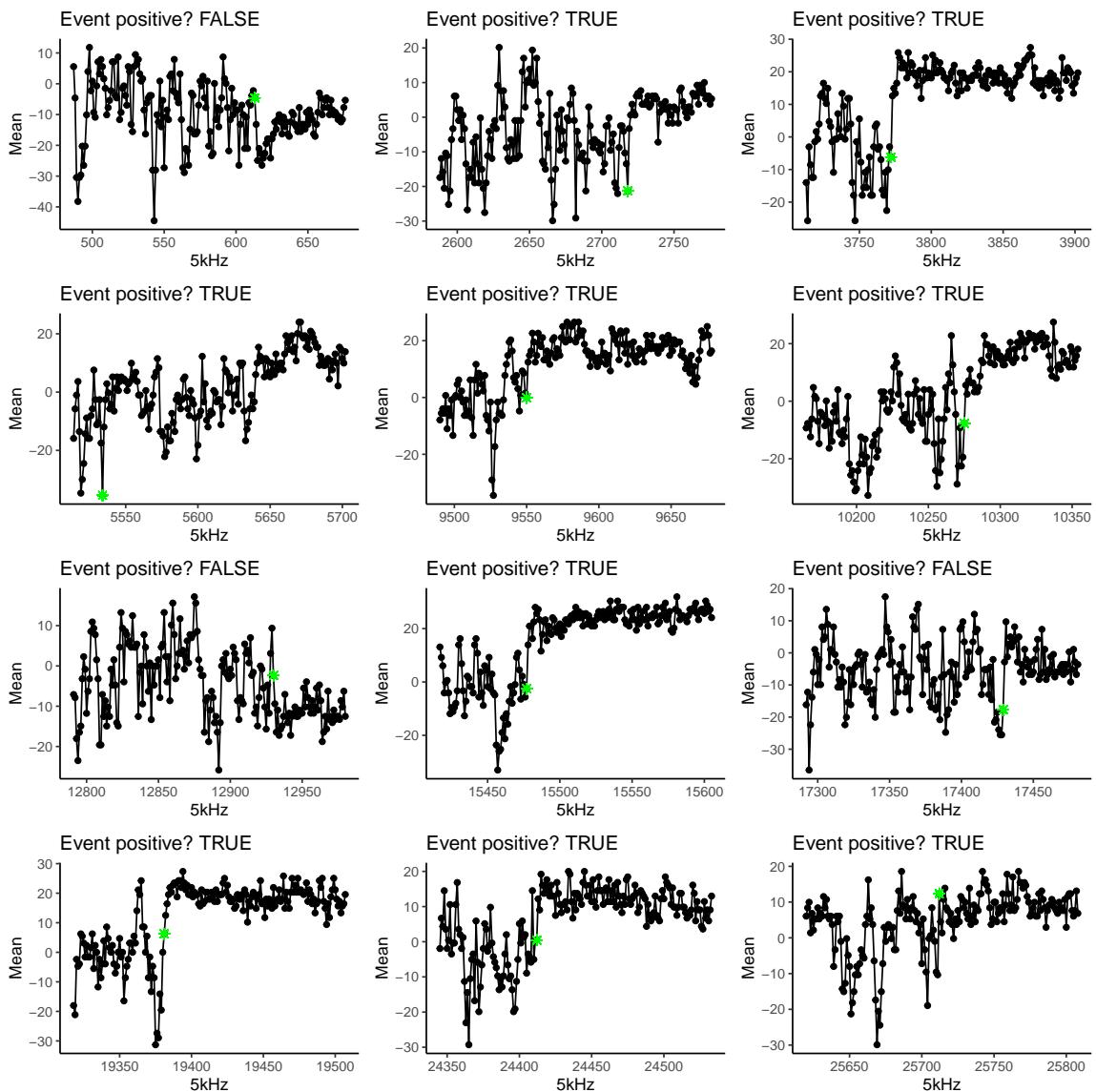


Figure 4.23. Changepoint identified start of event in the original data set

This same approach can be applied to the backside. Once backside change points are identified than more precise measurements of time on, time off, displacements, and forces can be estimated. For now, we can go ahead and make one final plot showing the complete analysis overlayed on the raw/processed data. The pink shades indicates the start/end of the event. The vertical dashed lines are placed at the peak displacement of each event and labeled with the step size and event duration. No analysis is perfect, some shorter events are missed. These are usually running windows which have a high variance due to the window overlapping baseline and an event.

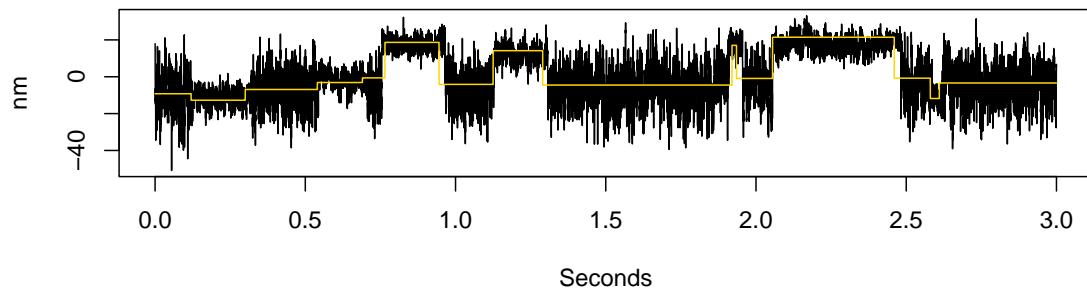


Figure 4.24. Results of analysis overlayed on data trace

4.3 User Manual

4.3.1 Getting Started

`{lasertrapr}` is an R-package/Shiny application built using the `{golem}` framework for automating the analysis of laser trap data. Please note that the app is currently still under development. Users should proceed with caution.

NOTE: an R-package is denoted by the {} braces

4.3.2 Install R & RStudio

Currently, `{lasertrapr}` can only be launched from an active R-session. Before you begin you will need to download and install R and RStudio. Both of these are free. RStudio is an IDE (integrated development environment) and is not 100% necessary, but is recommended. Follow the instructions on the respective websites to complete installation.

Additionally, you will need to install the R build toolchain. This allows a user to build an R-package from source on their own computer. The link provides directions on how to do this on Windows, Mac, and Linux.

At the moment there is a lot of overhead needed to get the application up and running. Hopefully, in the future a more stable version can be released onto CRAN (Comprehensive R Archive Network) or preferably as a standalone program.

4.3.3 Download `{lasertrapr}`

Open R/RStudio on your computer. You can download the latest stable version of the app from my *drat* repository on Github with the following code. Copy/paste the following into the R-Console or into an R-Script to run:

For users that want the latest developmental version to fork and contribute you can install from GitHub with:

Both `{devtools}` and `{lasertrapr}` will need to install dependencies. Update and install all the packages that they want when prompted. Alternatively you can install `{devtools}` from within the RStudio IDE by navigating to the “Packages” tab in the lower right box and clicking “Install”.

4.3.4 Launch the App

Once you have successfully installed and built the `{lasertrapr}` package, you are ready for launch:

Once the initial setup and installation is completed, the above two lines is the only code that will need to be run each time you want to use the app. You can update to the current developers version anytime by re-running `install.packages("lasertrapr")` or `devtools::install_github("brentscott93/lasertrapr")`

4.3.5 Upload Data

4.3.5.1 The `~/lasertrapr/` folder

The `{lasertrapr}` app offers more than just the analysis of single laser trap data traces, but the application also serves as an *opinionated data management* tool. The app forces users to adopt a specific directory tree structure. All data can be organized into specific **project**, **conditions**, **date**, and **observation** folders. Additionally, this structure can be leveraged in the post-analysis stage to easily handle statistical analyses and auto-generation of plots for quick summary features provided by the app. Upon initial launch of the app, a *lasertrapr* folder will be created under `~/lasertrapr`. The exact location of the folder will vary depending on OS, but you can get the exact location by running `path.expand("~/lasertrapr")` in the R-console. All of your data and analyses will be created and saved within this folder.

The app will take any uploaded data and copy it into an **observation** folder. In the laser trap, we will assume an “observation” is all the data collected at one mogul at a given time. This data, or observation, would have been collected on a specific day (date) under specific solution conditions (ionic strength, pH, mutations, etc.) and would belong to one bigger project. This logic provides the basis for the data management provided by the app. **The project, conditions, and date folders must exist in order for data to be uploaded.** The app will automatically create the observation folders when data is uploaded.

4.3.5.2 Create Folders

Located in the top right of the app is the **Folder Manager**. Click to open the menu (there is a known bug that when opening menu for first time some of the menu is cut off, you can click outside menu to close, and just re-opening will fix this menu display). Click the dropdown and either select your specific project folder or select “Create New...” to create a new folder. Avoid using spaces in the folder names. The prefix “project_” will be appended to the beginning. Continue to create folders in the same way for the conditions and date.

In lieu of using spaces, users are encouraged to use “-” and “under_score” in your conditions naming conventions. I prefer to use a combination of both. I use a “-” as a space *within* a given variable name and “under_score” as a space separator *between* distinct variables in the conditions. For instance, if I collected data with a wild-type myosin (WT) at pH 7.0 and 30mM Pi my conditions would be “WT_pH-7.0_30mM-P_i”. The app will add a “conditions” column to all the uploaded trapping data to ID each observation from one another (along with columns for the project, date, and observation info). The benefit of having a standard naming convention and knowing what the roles of the special space separators are will allow for more flexibility and robust analysis later on. For example, eventually the conditions column can be separated into *many* variable columns easily in the post-analysis by knowing that variables are separated by “under_score”. It doesn’t matter what you use, just be consistent.

4.3.5.3 Load Data

4.3.5.3.1 Simple Upload A simple upload means the user has a complete file of trap data. One file = one complete record.

Any regular delimited file type can be loaded into the app (thanks to `fread()` from `{data.table}`). Currently, the app only supports single trap detector data which

means only one signal columns can be used for analysis. **The first column in the data file will automatically be chosen as the trap data.** The app will make a copy of the data and re-format it for use within itself. Uploading a dataset to the app will create an “obs-##” folder inside the currently selected data folder. The data will be saved within that “obs-##” as “trap-data.csv”. This last detail is purely informative. One of the benefit of using the app is not worrying about file management. You technically will never need to look inside the files created by the app, but it is still nice to know how it works and where to find things.

Multiple files can be uploaded at once too. Each file will be treated as a new observation. If three files are uploaded, then the folders “obs-01”, “obs-02”, and “obs-03” will be created within the selected date folder and “trap-data.csv” files will be initialized from the uploaded data within the respective observation folders.

If you are uploading simulated data or data that has already been processed you can check the “Ready for Analysis?” box which will let you enter a trap stiffness (pN/nm) value and assumes you don’t need to convert data from mV to nm (the app will use a value of 1 for the mV to nm conversion during subsequent analysis). This allows users to skip the “Clean & Process” section of the app since the data will be ready for analysis once the data is initialized.

4.3.5.3.2 Split Observations This is a special use case for the Debold Lab. The trapping computer saves a separate .txt file for every 5-seconds of data collected. All .txt files from a given conditions/date combination can be uploaded at once and the app will read the time-stamps and concatenate corresponding observation files together based upon the user selected time threshold.

4.3.6 Clean and Process

4.3.6.1 Clean

Cleaning data *use* to be a painful and tedious task for me. It used to involve visualizing data traces in one software (pCLAMP), while jotting down notes in spreadsheets with specific time-stamps and then needing to convert time (in seconds) back to the original sampling frequency so I could go and manually find that data in excel to select/delete by hand (UGH! - I still cringe thinking of doing this...). Sometimes it still amazes me how much faster and easier this is with `{lasertrapr}`. The benefit of `{lasertrapr}` is even if you do not like/need the analyzers or other features you can easily just use it to clean, process, and export your data.

4.3.6.1.1 Cut data The most common use case for needing to cut data from a trap data trace is when during collection an actin-filament snaps or the myosin sticks down. In these cases, there is still good/usable data present in the trace, but the presence of the large signal disruptions caused by the snapping filament or stick-down could throw off the analyzers. The easiest fix is to cut these portion of the data out. I generally do **not** recommend deleting data except for these cases in which case I refer to this as “trimming” the data.

Trimming (deleting) an observation to make it analyzer ready is easy with `{lasertrapr}`. Use the **Folder Manager** to select an observation, load the observation, select the data to delete, and hit the Cut button. *NOTE: This permanently deletes the range of data selected from the trap trace and is irreversible (unless you re-upload your data).*

4.3.6.1.2 Move data In some cases, you do not want to delete data, but to split one record into 2 different observations. This is called “moving” data in `{lasertrapr}`. Sometimes, during collection stage drift occurs so the trace starts with a stable/horizontal time-series, but then over time the data starts to trend with

time upwards in the y-dimensions turning the signal into a diagonal line. One way to deal with this is to split the single **obs** into two separate ones so the two-halves can be processed separately with the diagonal portion getting detrended later.

Moving data is the same procedure as cutting data, except for the final button pushed! Load an **obs**, select the data to move, and click move. A new observation folder will be made with the selected data and the selected data will be deleted out of the current **obs**. *NOTE: This cannot be undone without manual intervention (you would have to load the trap-data.csv files into R and rbind them back together or re-upload the data and start again).*

4.3.6.2 Process

Another benefit that I have enjoyed while analyzing my own data with `{lasertrapr}` is the ability to easily visualize how processing will transform my data before deciding to save/analyze it. Currently, you can convert data from mV-to-nm with a pre-determined user conversion value, center the baseline mean to zero using either the “baseline range” or “remove mv” techniques, or you can detrend your data with a piece-wise linear detrend-er.

4.3.6.2.1 Convert to nm Short and sweet. Enter your pre-determined mV-to-nm conversion in the **Step Cal** box and hit **Graph** to preview.

4.3.6.2.2 Remove baseline When collecting laser trap data the detector is measuring the relative intensity of light across its four-quadrants. The data is saved in units of millivolts (mV) and is usually not centered around 0mV. So, when the data is converted to nanometers the y-axis range becomes some arbitrarily large or nonsense negative value. Technically there is nothing wrong with this since we are interested in making *relative* measurements of displacements from baseline, but it makes more sense and is easier to read when the y-axis is centered around 0nm. This can be

accomplished by calculating the average position of the baseline signal and subtracting that value from every point in the y-dimension. Baseline removal is currently implemented in 3 ways: baseline range, remove MV, and detrending the data.

4.3.6.2.3 Range The baseline range is simplest and the most “legacy” (i.e. this was easiest for me to implement when I was doing this all manually before `{lasertrapr}`). You can manually select a quiescent period of data that represents the baseline signal and the mean position of this period of data will be calculated. By selecting **remove base** from the **Graph Options** and hitting **Graph** to update the app will provide a preview. *NOTE: this will not be saved until you explicitly hit save.*

4.3.6.2.4 Remove MV Sometimes it can be tricky to find a nice quiescent period of baseline signal to calculate the range. This is especially the case with fast motors and mini-ensemble experiments. Instead it can be helpful to use the **Remove MV** option. This will perform a Mean-Variance transformation of the entire data trace and show the plot in an interactive window. You can then select the area that represents the baseline population, the mean is calculated, and by selecting **Remove MV** in **Graph Options** and hitting **Graph** to update, the app will provide a preview. *NOTE: this will not be saved until you explicitly hit save.*

4.3.6.2.5 Detrend data Stage drift can occur in longer records, or put another way the displacement on the y-axis will start to trend with time on the x-axis. There should be no relationship between time and displacement (slope should be 0). If this occurs the data record will look like it is tilted diagonally. This can be compensated by *de-trending* the data. A piecewise linear regression is fit to every 5 seconds of data and the resulting slope is removed from the data. Select **Detrend** in **Graph Options** and click **Graph** to preview the results. This also centers the baseline around 0. *NOTE: this will not be saved until you explicitly hit save.*

Your browser does not support the video tag.

4.3.6.3 To Include, or not to Include...

I do not like deleting data, but I also do not like wasting my time. Unfortunately, not all that glitters is gold, or not all trap data that is collected is usable. If I **know** that data does not look like exceptional signal-to-noise, there are no events, or will probably not analyze well I want to exclude those events from analysis so they do not take time getting analyzed etc.

By default, `{lasertrapr}` excludes all data from analysis so you need to **Include** the data for the app to analyze it.

If you like the data check the **Include** button when saving data.

4.3.6.4 Save!

NOTE: The app will not save anything unless you save the changes!

4.3.7 Analyzers

Currently, there are two analyzers available within the app. The HM-Model/Changepoint analyzer was designed for single molecule trapping data and the mini-ensemble analyzer is self-explanatory. Both analyzers have a similar UI with the ability to impose user control over some of the analysis parameters.

For both analyzers you have the option to run the analyzer through all of the observations in a given date folder, or you can select just a single observation. If you select to analyze a single observation you must also select the observation you want to analyze in the Folder Navigator (top right button).

4.3.7.1 HM-Model/Changepoint

Intended for single molecule data, this analyzer uses a combination approach to identify single molecule binding events. A Hidden Markov Model is implemented

with the {depmixS4} R-package on a running window transformation of the data to estimate locations of binding events. Then a changepoint analysis is applied to a small subset of the original trapping data around the HM-Model estimated transition periods with the {changepoint} R-package to precisely choose the most probable data point (at the original sampling frequency) where the binding event occurred.

Within the HM-Model/Changepoint analyzer, users have control (to some extent) over most aspects of the analyzer including the running window transformations, some of the HM-Model, and the type of changepoint.

Clicking the “Options” button will open up a menu that will allow you to set the analysis option.

4.3.7.1.1 Options: HM-Model The Hidden Markov Model analyzes a running window transformation of the original data trace. Here you can select the window width (in data points) and how you would like the window to progress. Users are referenced to Smith, Steffon, Simmons, and Sleep 2001 for further details on how to optimize the windows. Per their recommendations the default of the progression of the windows is by 1/2 overlap. **Note: not all window slide options have been tested. Possible bugs may exist and app crash potential.** You can just restart the app and try another option.

The “Channels” options lets you choose if you want the Hidden Markov Model to use both the running mean AND running variance transformations or just the running variance. A personal anecdotal recommendation is to use both the running mean and variance.

“EM Random Start” is FALSE (unchecked) by default. If TRUE (checked) the analyzer will use random number generation to start the EM-Algorithm.

4.3.7.1.2 Options: Changepoint There are two sections the changepoint options. Since changepoint analysis is applied separately to the beginning and ends of

the events so you can control the behavior of both. The default is to use the changepoint method “Mean/Var” which has the changepoint algorithm use the mean and variance position to identify the most probable change. Whereas if “Variance” is selected a slider will appear allowing the user to select a window width for the running variance transformation. The changepoint will then look for a change in the mean signal position of the variance transformation to identify the most probable change.

4.3.7.1.3 Options: Displacements Users can select one of two methods for peak displacements to be calculated. The “average” method calculates the mean signal position of the entire ID’d event minus the first and last 5ms. Alternatively, users can opt to use the “peak” method which returns a maximum value from a 5ms running mean of the ID’d event.

4.3.7.1.4 Options: Hz Users need to specify the sampling frequency (in Hz) for proper conversion between data points and seconds. Defaults to 5000 Hz.

4.3.7.2 Mini-Ensemble

This mini-ensemble analyzer uses a simple thresholding method to ID events. Users can control the threshold parameters for the displacement and minimum time on as well as the running window width.

4.3.8 Summarize

The summarize feature is designed to be more of a quick summary feature that provides an quick look at the summary statistics of your project with minimal user-implemented choices. I found it to be convenient when you just want a quick check on how a project is progressing wanting to see averages, standard errors, sample size, etc to update myself and the lab. You can change colors or re-reorder the factors for display purposes, but other than that it will create the same default statistical figures powered by the `{ggstatsplot}` package. This provides a quick statistical tasks

to check on significance. Currently, conditions are grouped individually and performed in a “one-way” fashion. Click the link to visit the `{ggstatsplot}` website for insight into the meanings of all the statistical symbols. These quick summaries can be exported to standalone .html dashboards to share. In the future I would like to add support to export to .ppt files as well.

Summarizing data will read, filter, combine, and save all “measured-events.csv” into your “summary” folder within your project with the date, project name, and “all-measured-events.csv” as the identifying file name. The summarized data will also be saved in a similar fashion but as “summary-data.csv”.

Note: The **split conditions** feature can be used to separate your **conditions** name into multiple unique variable ID’s which can be useful for later use when creating plots. Split conditions only works if you follow the condition naming convention described in these docs. No spaces - EVER! Underscores “`_`” separate distinct variables and dashes “`-`” can be used as spaces within a variable. When selecting split conditions, n number of textboxes will appear for the number of variables present in your conditions name, which is solved by identifying the number of underscores present plus 1. You can then enter the variables names which will become column names in the data.

4.3.9 Ensemble Average

Ensemble averaging in `{lasertrapr}` is divided into three distinct steps 1) preparing the ensembles, 2) averaging and fitting the ensembles, and 3) plotting. Separating these distinct tasks provides the user with more control over the process without having to re-run computationally long tasks repeatedly.

4.3.9.1 Prep Ensembles

An individual single molecule (myosin) event should be composed of two substeps (d1 & d2). Events can be aligned at the changepoint identified start or end of the

events, temporally synchronized to the same duration, and then averaged together to create one “average” event. The user can control how the event synchronization occurs.

All the original trapping data traces are saved in the “trap-data.csv” files and their ID’d events are saved in “measured-events.csv”. The “measured-events.csv” file not only includes displacements, event durations, etc., but also the changepoint ID’d datapoint for the start and end of each event. This information can be used for the ensemble averaging.

Ultimately, the “Prep Ensembles” button will create an “ensemble-data.csv” file in each **obs-##** folder that will have all of the individual forwards and backwards extended events that was constructed with the selected parameters described below in one file in long format.

This step usually takes the most time computationally as it involved reading in every single “trap-data.csv” file, extracting the events, and writing the synchronized events back to the folder.

NOTE: Only observations that were **included**, analyzed as **success**, and whose **review** marked as TRUE will be included in the ensemble averages. Additionally, if the observation passes that first check then any events that were user excluded will be filtered out. Each time you re-prep your ensembles, all existing “ensemble-data.csv” files are **ERASED** and the data is re-read and prepped again. This allows the user to go back and exclude a trace or event from analysis and re-prep without leaving old straggling files behind.

4.3.9.1.1 Extending Forward The goal with the forward ensemble average is to align the beginning of the events and “stretch” the ends of the short events to be equal to the longest event in time. The app lets the user choose how much (or little) time to use to extend the event. A 3ms “Avg of ms to extend forward” (I should probably come up with better names for these parameters) value means the position

of the last 3ms of an events will be averaged and that resulting average value will be used (repeated) to extend the event out. The window size you select would probably be dependent on [ATP]. The higher the [ATP] the lower the window should be to decrease the likelihood of averaging pre-hitch (d1) displacement position. Whereas, at high [ATP] myosin will spend a longer time in a post-hitch (d2) final displacement position waiting for ATP to induce dissociation so you can be less aggressive in the size of the window in attempts to truly capture the average of that final d2 displacement.

4.3.9.1.2 Extending Backwards Similar idea of the forwards, but for the backward ensemble average the goal is to capture the average position signal of the d1 pre-hitch position signal to extend the events. However, this is a little tricky. The changepoint algorithm used identifies the transition period going from un-bound to bound as a part of the event. This is the very brief period in time where the bead is moving through solution as it is displaced from the center of the trap. As a result, even though these datapoint are considered part of the event, they are not a true representation of the post-powerstroke/pre-hitch d1 position. The app gives the option to “skip” into the event before performing an average in effort to try to avoid averaging in these transition points. The “Number of ms to skip before s1 avg” (again not a catchy name...) lets the user decide how many ms to skip into each event before averaging. Similarly, the “Avg of ms to extend s1 backwards” allows the user to select how many ms to include in the average after skipping ahead.

4.3.9.2 Average & Fit Ensembles

Once the ensembled are prepped they can then be averaged and optionally fit with an exponential curve. The options allow the user to select the maximum amount of data to plot and whether to fit a single, double, or no exponential to the data. This part is relateively strighforward, just click “Avg Ensembles”. Averaging the ensmbles will automatically save the fit parameters under “lasertrapr/project/summary”.

4.3.9.3 Plot Ensembles

After the ensembles are averaged and optionally fit, you can navigate to the “Plot Options” tab to activate the graph. Prior to activating this tab only the Forward/Backwards Fit Parameter tables will be viewable.

Within the app, there is a limited selection of plot customizations that allow the user to tweak the appearance of the plot. You can provide custom labels to the facets, change facetting directions, shift the backwards ensembls underneath the forwards to save on space, change theme size, along with some other minor features. Plots can be saved with “Save Plot” and figures are saved under the selected project folder inside “lasertrapr/project/summary/figures”.

NOTE: In order to get the custom labels to work you must select a factor order.

4.4 Lasertrapr Cloud: Beta

The `{lasertrapr}` app personally solved many of my problems and now serves as my daily analysis application that provides automatic and reproducible analysis for my data collection as a laser trapper in grad school. However, one thing I kept noticing was that all the data I collected was stored locally on *my* computer and that no-one else had access to it. Moreover, not everyone in my lab has R installed let alone `{lasertrapr}`. I started to want a way to share my project so my lab members could see my data and stay up to date without needing to sit down with me on *my* computer just so I could scroll through the analyzed data traces to keep them up to date on a project. So I began to think and realized, the `{lasertrapr}` app is built using the `{shiny}` R-package which is a web-framework for building apps in R. The GUI portion is HTML/CSS and the back-end server code is executed in R. Usually, a typical `{lasertrapr}` user would install the app from within an active R-session that is run on their person computer, load the `{lasertrapr}` library, and run the app locally on their computer. Running the app locally on a users computer has the benefits of using

their personal computer’s hardware resources and keeps the data on their hard drive, but the same application framework would also run on *someone else’s computer* (i.e. a server/“the cloud”). Here, is a beta test of a **{lasertrapr}** extension called *Lasertrapr Cloud*. With *Lasertrapr Cloud* a user can archive one of their local **{lasertrapr}** projects to Google Drive and push an instance of “Lasertapr Cloud* to a remote server that is accessible via URL which essentially provides a web-based viewing portal to the archived data. This means that simply anyone with a web-browser can navigate to the URL, enter the password, download, and view the analyzed trap data and current project summary along with providing a partial backup of data. For the *Lasertrapr Cloud* viewer to work first a project needs to be archived and backed up to a shared google drive (this is all automated by the app). Since this involved large file transfer the files are compressed locally (.zip) before being uploaded/downloaded as the smaller file sizes have faster transfer speeds. When a visitor arrives at the URL for the cloud viewer and enters the correct password they are able to select an available project and compute the summary statistics or view the analyzed data which downloads the compressed data and displays the data in the same widgets available in the regular desktop app to view the”overlaid” analysis results ontop of the raw data. At the time of writing, I have some of the data I collected for this dissertation available to view at <https://debold-lab.shinyapps.io/lasertrapr-cloud/>. The password is “tr@p”. Once inside the app you can select from either of 2 archived projects to explore the myosin V data I collected for Chapter 5. At the moment, I am not yet as to whether *Lasertrapr Cloud* will continue to be developed or even if this section will stay in my dissertation.

CHAPTER 5

MYOSIN'S POWERSTROKE OCCURS PRIOR TO THE RELEASE OF PHOSPHATE FROM THE ACTIVE SITE

This chapter of the dissertation has been published in *Cytoskeleton* (Scott et al. 2021) and can be read in the journal's formatted PDF version. The following chapter is a re-formatted version of the published article, but contains additional supplemental sections to expound upon the material not included in the publication. The full citation for the published article is “Scott, B., Marang, C., Woodward, M., & Debolt, E. P. (2021). Myosin’s powerstroke occurs prior to the release of phosphate from the active site. *Cytoskeleton*, 1–14. <https://doi.org/10.1002/cm.21682>”

5.1 Introduction

Myosins are a family of motor proteins responsible for generating force and/or motion inside the cell (Foth, Goedecke, & Soldati, 2006). Muscle myosin II is the most well-characterized class and is the molecular motor that drives muscle contraction (Geeves & Holmes, 1999). Myosin Va has a motor domain that is highly homologous to myosin II (Foth et al., 2006) but its task is the intracellular transport of vesicles, walking processively along the 36 nm pseudo-repeat of actin filaments (Warshaw et al., 2005; Yildiz et al., 2003). Other forms of myosin and related molecular motors (e.g., kinesin) play important roles in intracellular transport, cell migration, the maintenance of cell structure, and even in mitosis (Foth et al., 2006).

Common to all myosins is the ability to convert the chemical energy of ATP into mechanical work, however key molecular details of this process remain unclear

(Houdusse & Sweeney, 2016; Takagi, Homsher, Goldman, & Shuman, 2006). In a precisely coordinated sequence of steps the ATP is hydrolyzed off of actin and then the products are released while it is tightly bound to an actin filament (Holmes & Geeves, 2000; Sweeney & Houdusse, 2010). In a simple model of the cross-bridge cycle, ATP enters the active site with myosin tightly bound to actin in the apo or rigor state. This induces conformational changes that cause myosin to dissociate from actin (Geeves & Holmes, 2005). ATP is then hydrolyzed to ADP and P_i off of actin, but the products remain in the active site while myosin is detached from actin. The re-binding to actin triggers the release of P_i and then ADP (Bagshaw & Trentham, 1974). The powerstroke is thought to occur close in time to the release of P_i (Holmes & Geeves, 2000; Houdusse & Sweeney, 2016; Sweeney & Houdusse, 2010; Takagi, Shuman, & Goldman, 2004) from the active site, however the exact sequence of events is unclear and is currently the source of a vigorous debate within the field (Gunther et al., 2020; Llinas et al., 2015; Muretta, Rohde, Johnsrud, Cornea, & Thomas, 2015; Trivedi et al., 2015; Woody, Winkelmann, Capitanio, Ostap, & Goldman, 2019). The key unanswered question is which event occurs first the powerstroke or the release of P_i from the active site (Llinas et al., 2015; Woody et al., 2019)? In other words, does the mechanical event “gate” the biochemical changes, or do the chemical reactions “gate” the mechanical events? The answer to this question has important implications for the long-standing pursuit of the molecular basis of force-generation by muscle (Huxley & Simmons, 1971; Linari et al., 2015; Piazzesi et al., 2002; Veigel, Molloy, Schmitz, & Kendrick-Jones, 2003), but it also has broader implications for understanding the fundamental nature of energy transduction by nucleotide-based motor proteins (Vale, 1996, 2003).

The timing of force-generation relative to P_i-release was first addressed in single skeletal muscle fibers (myosin II); these studies demonstrated that the development of force preceded the release of the first hydrolysis product, P_i (He et al., 1997; Sleep,

Irving, & Burton, 2005). Consistent with these observations, the rapid release of caged-P_i demonstrated a clear delay in the depression of isometric force in muscle fibers (Dantzig, Goldman, Millar, Lacktis, & Homsher, 1992). These findings were consistent with force-generation preceding the release of P_i; however, the spatial and temporal resolution of these assays could not provide direct evidence of how and when a single myosin was progressing through its mechanical or biochemical transitions. More molecular level detail was provided by work using Förster Resonance Energy Transfer (FRET) probes on isolated myosin molecules in bulk solution assays, to directly monitor the position of the lever arm while also tracking the rate of P_i-release using a rapid-reporting, phosphate-binding-protein (Muretta et al., 2015; Trivedi et al., 2015). These assays demonstrated that lever arm rotation occurred rapidly after myosin strongly bound to actin, at a rate of $350. \text{ s}^{-1}$, while P_i-release from the strongly bound state occurred at $\sim 30. \text{ s}^{-1}$, suggesting that P_i-release occurs after lever arm rotation for myosin II. Similar findings were observed using myosin Va with the powerstroke occurring at $400. \text{ s}^{-1}$, while P_i-release occurred at $200. \text{ s}^{-1}$ (Trivedi et al., 2015). Similarly, single molecule observations, with microsecond time resolution, suggest that in cardiac myosin lever arm rotation may occur even faster, at $1000\text{--}5000. \text{ s}^{-1}$ (Woody et al., 2019), exceeding all estimates of P_i-release from myosin II by at least an order of magnitude (Sleep et al., 2005). Thus, the findings from these types of functional assays support the hypothesis that the powerstroke occurs prior to P_i-release.

However, x-ray crystallography structures of myosin, captured in various states of the cross-bridge cycle, suggest that P_i-release must occur before lever arm rotation (Llinas et al., 2015; Sweeney & Houdusse, 2010). Specifically, structures of myosin trapped in different nucleotide states using various nucleotide analogs show that the lever arm exists in a pre-powerstroke position when the gamma-P_i (or an equivalent analog) is still in the active site, and only achieves a post-powerstroke position when

the gamma-P_i is absent from the active site (Robert-Paganin et al., 2020). Based on these and similar structural observations it was hypothesized that upon binding to actin, conformational changes in myosin's active site occur that allow P_i to be released from the active site, and only then can the conformational changes that lead to lever arm rotation occur. Thus, this hypothesis posits that P_i-release "gates" the lever arm rotation, therefore the lever arm cannot rotate (i.e., the powerstroke cannot occur) until P_i has left the active site (Gulick et al., 2000; Rayment et al., 1993a).

In attempt to resolve the conflict between the functional and structural evidence, Llinas et al. (Llinas et al., 2015) formed myosin crystals soaked in elevated levels of P_i for varying durations before rapidly freezing the samples and performing x-ray crystallography. With a short exposure time before freezing, P_i was observed to be either at the exit of the escape tunnel adjacent to the active site (P_iR1) or close to ADP in the nucleotide binding region (P_i R2, that is, still in the active site). Longer delays before freezing revealed that P_i diffused back into the active site close to ADP, and when it did, the lever arm returned to the prepowerstroke position. These observations led the authors to hypothesize that P_i leaves the active site very rapidly (and prior to the powerstroke) but stays in the exit tunnel before it is released into solution. Thus P_i-release from the active site may occur much more rapidly than P_i appears in solution. This idea could therefore potentially explain why functional assays observe a rate of P_i-release that is slower than the rate of force development and the powerstroke (Muretta et al., 2015; Trivedi et al., 2015). To further test this hypothesis, Llinas et al. (Llinas et al., 2015) introduced a mutation into switch I that was designed to slow P_i-release; by impeding it from entering the exit tunnel (S217A in myosin Va). The loss of the hydroxyl group, thought to make contact with the gamma phosphate of ATP (Forgacs et al., 2009; Smith & Rayment, 1996), is hypothesized to impede the entry of P_i into the exit tunnel (Llinas et al., 2015). Consistent with this hypothesis, actin-activated P_i-release is

3–10-fold slower in this mutation compared to WT (Forgacs et al., 2009; Gunther et al., 2020; Llinas et al., 2015). In addition, x-ray crystal structures of myosin soaked in high concentrations of P_i for 45 min show P_i in the active site near ADP with myosin occupying a prepowerstroke state, supporting a P_i -release gated powerstroke (Llinas et al., 2015). Thus, the release of P_i may occur faster than the lever arm rotation, but it may not appear in solution immediately because it is still in the P_i exit tunnel. If correct, this would provide an explanation for the discrepancy in the results between structural and functional findings. Indeed, it was recently proposed that this structural information provides a hypothesis that unifies prior findings from solution and functional experiments (Robert-Paganin et al., 2020).

This hypothesis is also important because it makes specific and testable predictions about how a single myosin molecule should behave if P_i is maintained in its active site. For example, because P_i gates the powerstroke in this model, P_i should only rebind to actomyosin in a prepowerstroke state because P_i can only leave or re-enter the active site when the lever arm is in a prepowerstroke position (Llinas et al., 2015; Sweeney & Houdusse, 2010). Indeed, in this model, the rebinding of P_i to the active site prevents the powerstroke from occurring rather than reversing it. By exposing myosin to elevated levels of P_i to maintain P_i in the active site, lever arm rotation (i.e., powerstroke) generated by myosin after it binds to actin should be delayed or even prevented if P_i gates the powerstroke. Similarly, a myosin construct with a mutation that slows the entry of P_i into the P_i exit tunnel (e.g., S217A; Forgacs et al., 2009; Llinas et al., 2015), should also dramatically delay, or even prevent, myosin from generating a powerstroke once it strongly binds to actin. These effects would be most evident at the single molecule level where the generation of a powerstroke can be directly observed, and the duration of single actomyosin interactions directly quantified. Therefore, we directly tested this hypothesis using both approaches (high

levels of P_i and the S217A mutation) in a single molecule laser trap assay using a single-headed construct of myosin Va.

5.2 Results

To determine the order of the powerstroke relative to P_i -release from myosin's active site, we directly observed the powerstroke and strongly bound lifetime of a single-headed 1IQ construct of myosin Va using a three-bead single molecule laser trap assay (Figure 4.1a,b). In the first series of experiments, we examined the effect of 30 mM P_i on the size of myosin's powerstroke. To ensure that the elevated P_i would preferentially rebinding to an actomyosin state with ADP still in the active site (AM.ADP), the ATP concentration was maintained at 100 M for this set of experiments, a value well above the estimated k_m of 17 M (Baker et al., 2004; Forgacs et al., 2009). Single actomyosin interactions (Figure 4.1c) were detected using an algorithm based on a Hidden-Markov Model (Smith et al., 2001), and transitions into and out of single binding events were located using a Changepoint analysis technique (Blackwell et al., 2021; see Section 5).

The 1 IQ construct of myosin Va generated a 7 ± 0.6 nm (Mean \pm SEM) powerstroke (Figure 4.1b) that was unchanged by elevating P_i to 30 mM in the experimental solution (Figure 4.1e). The S217A construct also generated a step that was not statistically different from the WT myosin construct, either in the absence or presence of 30 mM P_i (Figure 4.1e). This suggests that when P_i -release is slowed from the active site, by either the S217A mutation (Figure 4.1b) or by P_i rebinding to the open active site, myosin generates the same size powerstroke upon binding to the actin filament.

Figure 4.1 shows in (a) The x-ray crystal structure of myosin Va (Shen et al., 2016), Images rendered in Swiss PDF viewer. (b) A schematic representation of the three-bead laser trap assays used. Myosin Va S1 with a single light-chain binding domain was adhered to the pedestal (3 m bead) via an anti-myoC antibody. Dis-

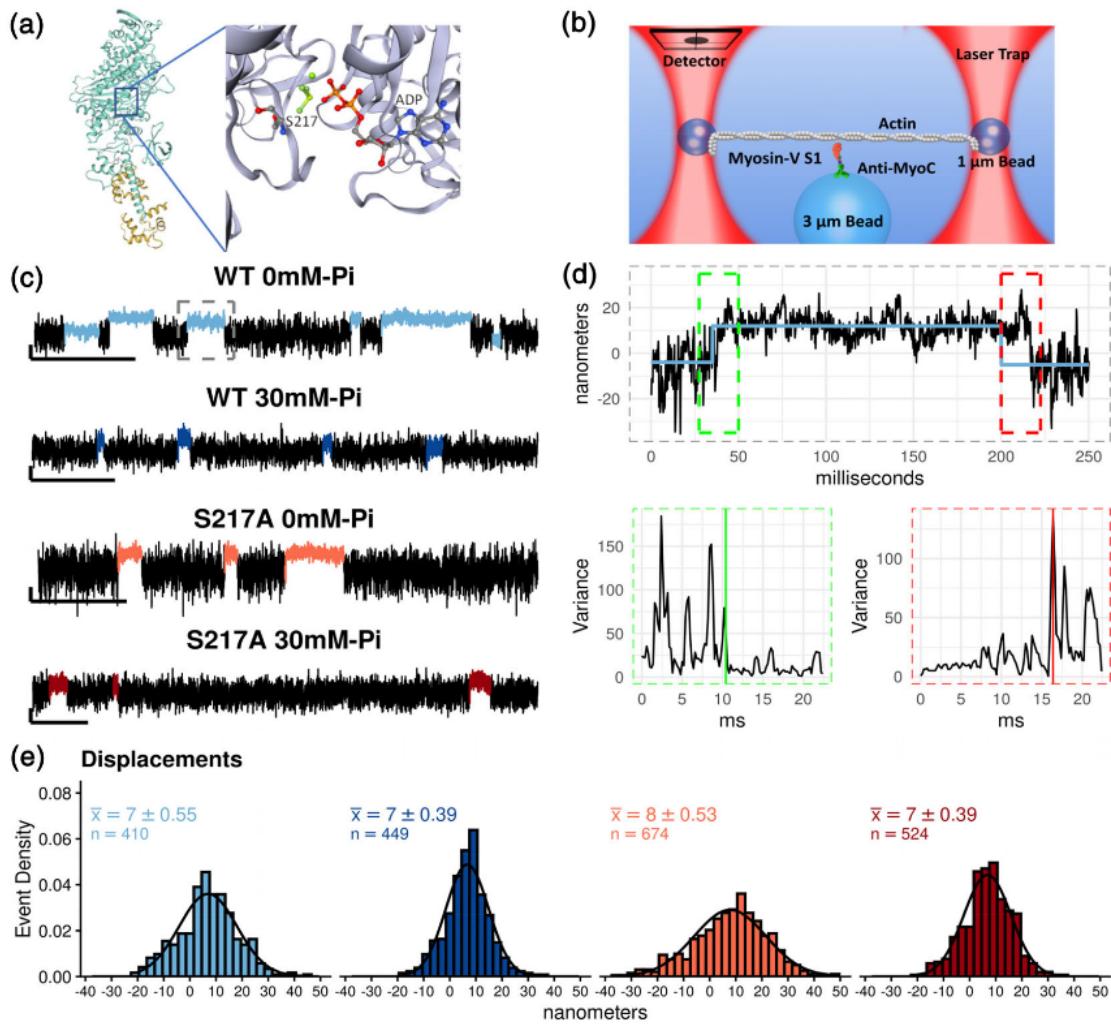


Figure 5.1. Single molecule laser trap assay data and analysis.

placements of the actin filament upon myosin binding were tracked using a quadrant-photodiode (detector). (c) Raw displacement traces for each myosin construct collected at 100 M ATP, and in the presence or absence of 30 mM P_i . Scale bar indicates 20 nm in the vertical axis and 500 ms in the horizontal axis. Actomyosin strong-binding events as identified by the analysis are colored differently than baseline noise for each condition and used consistently through the text. (d) Method of event detection shown for a binding event for WT myosin at 0 mM P_i . A single binding event in “c” (dashed gray box) is used to illustrate event detection. A Hidden–Markov Model was used to identify the approximate location of a binding event (blue line in upper graph) and Change-point analysis was used to precisely determine the start, and end, of the each event (dashed green and red boxes, upper graph) based on abrupt changes in the running variance of the signal both at the start and end of the binding event (lower graphs). See Section 5 for further details of binding event detection. (e) Histogram of binding events fit to Gaussian curves. The mean \pm SEM are shown with the sample size (n) indicated for both conditions and constructs. A 2-way ANOVA (myosin x P_i) analysis reported no significant ($p < .05$) differences.

To further address the relative timing of powerstroke and P_i release, we examined the rate of the transition from the unbound/weakly bound state into the strongly bound state (Figure 4.2a) using an ensemble averaging analysis previously described (Sellers & Veigel, 2010; Veigel et al., 2003), with minor modifications. In this analysis, the start and end of identified actomyosin binding events were temporally aligned (see Section 5). The front ends of the binding events were fit to a double exponential ($y = d1(1 - \exp[-x^* * k0]) + d2(1 - \exp[-x^* * k1])$) to estimate the transition rate ($k0$) from the detached/weakly attached states to the postpowerstroke state, of the primary powerstroke (Figure 4.2a) and a second rate ($k1$) reflecting the transition associated with the secondary powerstroke (see Section 5 for additional details). The initial rate, $k0$, was rapid and similar in the absence and presence of P_i for both constructs (604

and $467. \text{ s}^{-1}$ for WT at 0 and 30 mM P_i , and 597 and $757. \text{ s}^{-1}$ for S217A at 0 and 30 mM P_i , respectively). This rate (k_0) is thought to reflect several processes including the initial Brownian capture of the actin filament by myosin and the powerstroke (Blackwell et al., 2021; Veigel et al., 2003). Furthermore, the motion of the beads is damped by the viscous forces acting on the optically trapped 1- m diameter bead, with a corner frequency of ~ 500 Hz, which limits this rate (Neuman & Block, 2004). Despite these temporal constraints, if P_i -release occurred prior to the powerstroke, 30 mM P_i should have reduced this rate dramatically. This was not observed (Figure 4.2a).

To further investigate whether a pause occurred prior to the powerstroke, we examined the slope of the first 2 ms of the initial transitions in the ensemble averaged data (Figure 4.2e). During the first 2 ms of this transition, the powerstroke of myosin generates an average force of 4pN (Finer, Simmons, & Spudich, 1994), enabling it to move the damped bead more quickly than $500. \text{ s}^{-1}$ through solution, thus making it an even more sensitive measure for detecting a delay prior to the powerstroke. 2 ms was chosen because 90% of the powerstroke was completed within this time frame and a longer time window would include the plateau after the powerstroke was completed and thus would not reflect the time during the primary powerstroke (Figure 4.2). The slope of linear fits to these data show that for both constructs (WT and S217A), and conditions (0 and 30 mM P_i), the transition during the first 2 ms was extremely rapid and occurred immediately upon strongly binding to the actin filament (Figure 4.2e). This result strongly suggests that myosin generates the powerstroke immediately upon strongly binding to the actin filament and without a delay to allow for the release of P_i from the active site.

To confirm that our analyses could detect a delay in the generation of the powerstroke if it occurred, we performed simulations of single binding events in which myosin bound to actin and either, rapidly generated a powerstroke (Figure 4.2b) or

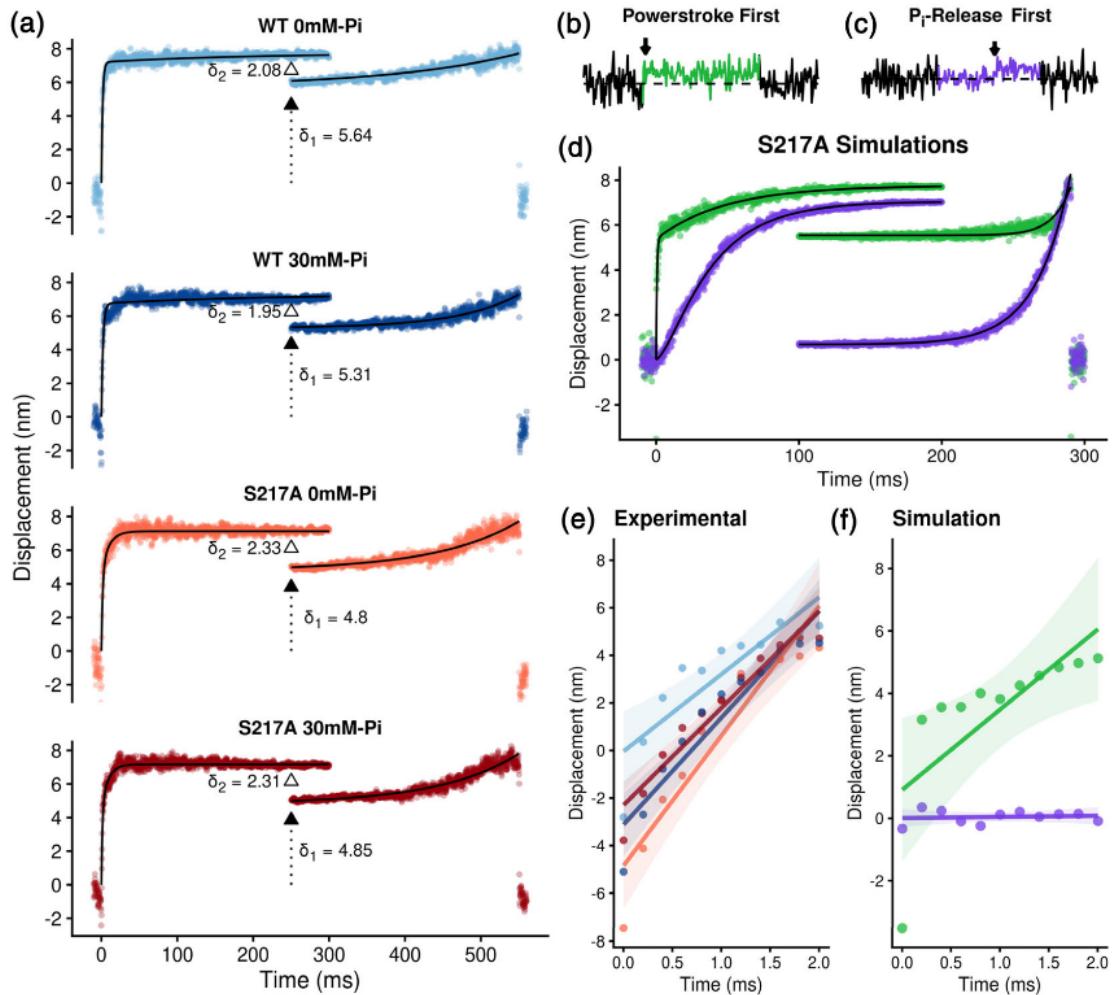


Figure 5.2. Ensemble averaged binding events from Scott Marang 2021.

paused with a time constant of 30 ms before generating a step (Figure 4.2c). The time constant for P_i -release was chosen to be consistent with the average measured rate of P_i -release from the S217A construct from three different reports (Forgacs et al., 2009; Gunther et al., 2020; Llinas et al., 2015). Analysis of these simulated data confirmed that such a pause would have appeared as a zero slope during the first 2 ms of the binding event (Figure 4.2f) and would have been visible as a slowed rate of transition into the strongly bound state in the ensemble average analysis (Figure 4.2d). In contrast to this prediction for a P_i -release first model, we observed a very rapid transition from bound to unbound in the experimental data in both constructs, as well as in the absence and presence of added P_i (Figure 4.2e). There was also no significant effect on the rates of transitions in the ensemble average analysis in the experimental data (Figure 4.2d vs. a). Collectively, these findings suggest that neither elevated P_i nor the S217A substitution slowed the rate of myosin's powerstroke, or induced a pause prior to generation of the powerstroke.

To expound upon the details for Figure 4.1, in (a) the events detected as indicated in Figure 4.1 were aligned and averaged for each condition and construct. Both the start and end of events were temporally aligned and the start fit with a double exponential fit: $y = d1 (1 - \exp[-k0 * x]) + d2 (1 - \exp[-k1 * x])$, where $d1$ is the displacement associated with the primary powerstroke and $d2$ the secondary powerstroke. $k0$ and $k1$ represent the rates of the transition from unbound to the postpowerstroke state and the transition to the secondary powerstroke state, respectively (see Section 5). Backward ensembles were fit with single exponentials of the form: $y = d1 + (d2 * \exp [k2 * x])$ (see Section 5). The first component of the fit to the start of the event reflects the transition from weak/unattached to a strongly bound postpowerstroke state. These rates were 614 and 467 s^{-1} for WT myosin Va in the absence and presence of 30 mM P_i , respectively and 596 and 757. s^{-1} for the S217A construct in the absence and presence of P_i , respectively. To determine the

size of the secondary powerstroke (2) the average of the last 100 data points of the front end of the ensemble averaged events were subtracted from the first 100 data points of the events aligned to the back end of the events (hollow triangle). This value was then subtracted from the total displacement to determine 1 (filled triangle with dotted line) using methods previously described (Capitanio et al., 2006). Under both conditions (0 and 30 mM P_i) and in both constructs (WT and S217A) 1 and 2 were not significantly different. (b) Simulated single molecule binding events using either a powerstroke-first model (green) or a P_i -release-first model (purple, c). Black arrows in b and c indicate the location of the powerstroke. Details of the simulations are provided in Figure S1 of the Supplementary Materials. The red arrow indicates when the powerstroke occurs during the binding event. (d) 500 binding events were simulated for each model and then ensemble averaged using the same procedure as the experimental data (see Section 5 for details). The P_i -release first data revealed a slower rate of transition into a binding event (104 vs. 1800 s^{-1} for the P_i -release- and powerstroke-first model, respectively). (e) An expanded view of the first 2 ms of the ensemble averages. The first 2 ms were used as the transition from unbound/weakly bound prepowerstroke to strongly bound postpowerstroke is >90% complete in this time frame. Data points were fit to a least-squares linear fit, with the shaded area representing a 95% C.I. for each line fit. Additional parameters of this analysis are displayed in Table S1. (f) Ensemble averaged data from simulations of S217A data using a powerstroke-first model (green) and a P_i -release-first model (purple)

While the elevated P_i did not affect the size of myosin's powerstroke in either construct, we also examined the duration of binding events for evidence that P_i rebound to myosin's active site. Indeed, in the WT construct, elevated P_i reduced the average lifetime (Figure 4.3a,b) suggesting that P_i rebound to actomyosin and accelerated myosin's detachment from actin. The difference in the arithmetic means was not significant, but the mean in the WT 0 mM- P_i condition was quite far from the me-

dian, indicating a skew in the distribution (Figure 4.3b). Indeed, this was because the decreased lifetime was not uniform across the distribution of binding event durations, rather it was due to a reduction in the number of long duration binding events, especially those >400 ms. For example, in the presence of P_i only, one binding event lasted longer than 1,000 ms (Figure 4.3b). In contrast, in the absence of P_i , 25% of the events lasted longer than 500 ms (Figure 4.3a, main graph). The effect of P_i was most evident in a quantile-quantile (Q-Q) plot of binding event durations versus the theoretical exponential fit (Figure 4.3c). And a Kolmogorov–Smirnov test revealed that there was a trend ($p = 0.06$) toward a difference between the distribution for the absence versus presence of 30 mM P_i . This finding suggests that P_i rebound to longer duration events in an AM.ADP state and accelerated detachment from actin. Elevated levels of P_i did not affect the event lifetimes of S217A, however in absence of P_i this construct has a shorter bound lifetime than the WT (Figure 4.3a,b), which likely reflects acceleration in the ADP-release rate caused by this substitution (Forgacs et al., 2009; Gunther et al., 2020). The accelerated rate of ADP-release may reduce its vulnerability to the rebinding of P_i , providing an explanation for the absence of a P_i -induced reduction in binding event durations (Figure 4.3b).

In Figure 4.3, (a) Cumulative distribution of the lifetime of the single molecule binding events. Color scheme same as in Figure 4.1. The y-axis is truncated below 70% of events to demonstrate the effect of P_i on lifetimes. (b) inset graph, histogram of the lifetime of all binding events for each condition (P_i) and each myosin construct. Dashed vertical line indicates the arithmetic mean and boxplot underneath displays outliers >1.5 times the interquartile range. A Kruskal–Wallis, nonparametric, ANOVA was used to test for significant ($p < .05$) differences among conditions, none were observed, however the distribution of events was altered (see Section 2). (c) A quantile-quantile (Q-Q) plot showing the location and magnitude of deviation of individual event durations (points) from a theoretical single exponential fit (solid

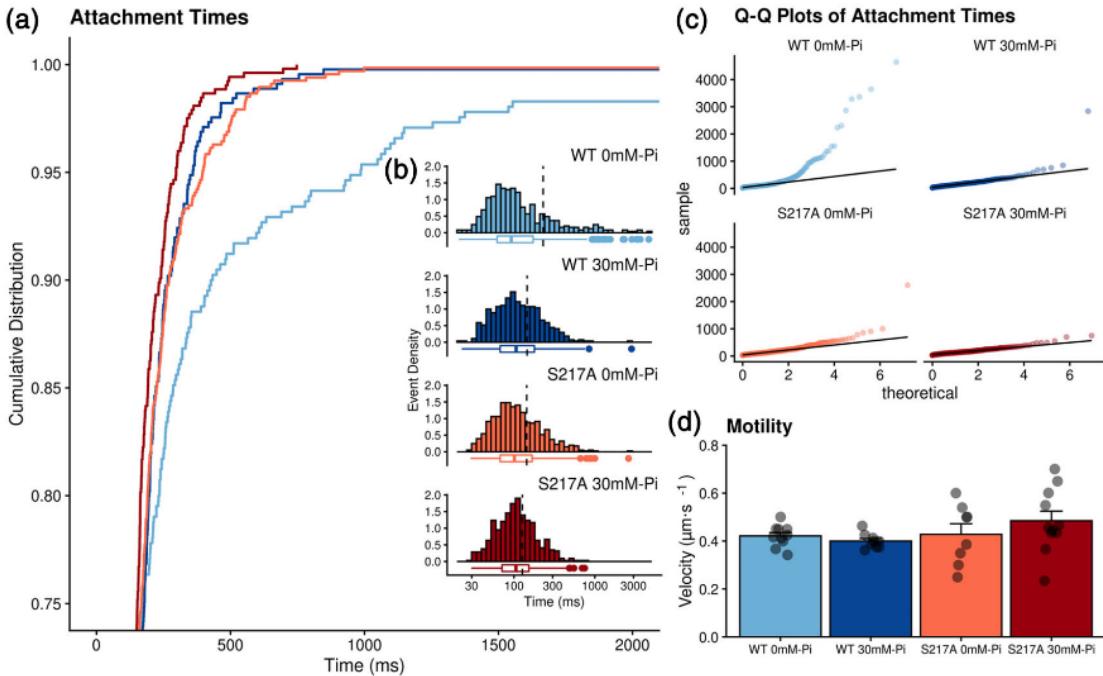


Figure 5.3. Binding event durations and bulk assays measurements.

black line). A comparison of the 0 versus 30 mM P_i distributions using a Kolmogorov–Smirnov, nonparametric test reveal a trend ($p = .06$) toward a difference. (d) Bar graph of Actin filament velocities from an in vitro motility assay performed for each condition and construct. Bar represents mean \pm SEM from eight experiments for each condition (points). Each data point represents the average velocity of three separate fields of view, that contained \sim 5–25 actin filament trajectories. Data were collected at 10 frames per second and the displacements were quantified frame by frame using MTrackJ (ImageJ plugin) as previously described (Debold, Turner, Stout, & Walcott, 2011). Data collected at 2 mM ATP at 125 mM total ionic strength. No significant ($p < .05$) differences were detected with a one-way ANOVA.

The frequency of binding events was determined by dividing the total number of binding events by the total amount of time collected. The values were 0.99 and $0.98. \text{ s}^{-1}$ for WT myosin Va in the absence and presence of 30 mM P_i, respectively; and 0.99 and $1.00. \text{ s}^{-1}$ for the S217A construct in the absence and presence of 30 mM

P_i , respectively. These findings are consistent with myosin's attachment rate being unaffected by elevated P_i or by the presence of the S217A mutation.

To determine the fate of the cross-bridge following the powerstroke and the rebinding of P_i , we again used the ensemble averaging analysis. A similar analysis has been used previously to detect and quantify the putative secondary powerstroke, or hitch, that is temporally associated with ADP-release by myosin (Sellers & Veigel, 2010; Veigel et al., 2003). We determined the size of the hitch by quantifying the difference in magnitude between the final displacements found at the ends of the forward ensembles and the displacement found at the beginning of the backwards ensembles, for each condition (Figure 4.2a). This analysis confirmed that WT myosin Va generates a hitch that was 25% of the total displacement (~2 nm) consistent with previous estimates using a myosin Va construct with a longer lever arm (Sellers & Veigel, 2010). Elevated levels of P_i did not affect the magnitude of the hitch, nor did the presence of the S217A mutation (Figure 4.2a). However, given that only the longest 25% of binding events were affected by P_i (Figure 4.3a,b) it is possible that the rebinding of P_i did occur prior to the hitch. This was also likely due to the limited amount of time spent in rigor at the final post-hitch displacement at the relatively high 100 M [ATP], a concentration needed to ensure that P_i primarily rebound to the AM.ADP state.

Previous reports suggest that the rebinding of P_i to actomyosin in the AM.ADP state can induce the reversal of the powerstroke, especially when the cross-bridge is strained (Woody et al., 2019). In our assay this would have appeared as displacement in the opposite direction to the initial powerstroke, which would have occurred sometime after the initial powerstroke but while myosin remained strongly bound to the actin filament. However, we did not see any evidence of this behavior either in the raw displacement records (Figure 4.1c), or in the ensemble average analysis (Figure

4.3a), where a reversal should have appeared as a downward slope in the transition out of a binding event, as others have observed (Woody et al., 2019).

It is possible that any reversal of the powerstroke and subsequent detachment from actin occurred faster than the resolution of our instrumentation (~ 2 ms). Therefore, we measured the effect of 30 mM P_i on the velocity of actin filaments in an in vitro motility assay (Vactin, Figure 4.3d); if P_i induced a reversal of the powerstroke we would expect that Vactin would be decreased by the presence of P_i in this assay. However, Vactin was unaffected by 30 mM P_i concentration (Figure 4.3c). Thus, under the present conditions, the rebinding of P_i does not appear to cause a reversal of the powerstroke in an S1 construct of myosin Va, despite accelerating the detachment rate in WT of the longest 25% of events (Figure 4.3a).

5.3 Discussion

5.3.1 The powerstroke precedes P_i -release from the active site

Functional evidence from both myosin II and myosin V suggests that the powerstroke occurs prior to P_i -release (Muretta et al., 2015; Sleep et al., 2005; Trivedi et al., 2015; Woody et al., 2019), while recent structural evidence from x-ray crystallography studies led to the notion that P_i -release must occur prior to the powerstroke (Houdusse & Sweeney, 2016; Llinas et al., 2015; Robert-Paganin et al., 2020). Based on new structural insights, it was proposed (Llinas et al., 2015) that this seemingly contradictory evidence could be resolved if P_i is released from the active site very rapidly but pauses temporarily in the P_i exit tunnel, before being released into solution. Structural evidence was provided to support this model, including the use of the S217A mutation in switch I which slows P_i -release from the active site (Forgacs et al., 2009; Llinas et al., 2015).

We directly tested this new model in two ways; first, by elevating P_i , and second by introducing a mutation that slows P_i -release from the active site. Using the WT

construct, we found that in the presence of 30 mM P_imyosin generated a powerstroke that was similar in size (Figure 4.1e) and rate (Figure 4.2e) to that observed in the absence of P_i. In this experiment, the elevated P_iin the buffer enables P_ito rebind to the active site, therefore if P_i-release gates the powerstroke, the P_iwould rebind to myosin prior to the generation of a powerstroke. This would have been evident at the single molecule level as a reduction in the size and rate of the powerstroke. However, no reductions in the size or the rate of the powerstroke were observed (Figures 1 and 2). Additionally, the altered distribution of strongly bound lifetimes (Figure 4.3a–c) provides evidence that P_i rebinds to myosin’s active site, because P_iis known to decrease myosin’s affinity for actin, thereby accelerating detachment (Geeves & Holmes, 2005; Holmes & Geeves, 2000; Takagi et al., 2004). Thus, the most likely scenario in this experiment (see blue arrows in Figure 4) is that myosin initially bound to the actin filament, and rapidly generated a powerstroke (500. s⁻¹ , see Figure 4.2a,e) that was similar in size to that observed in the absence of P_i (Figure 4.1e). Myosin then released P_i, at a rate of 150–200. s⁻¹ (Cruz, Wells, Rosenfeld, Ostap, & Sweeney, 1999; Forgacs et al., 2009; Llinas et al., 2015; Rosenfeld & Sweeney, 2004; Trivedi et al., 2015), but in 25% of the events a new P_ifrom solution quickly rebound in the active site while myosin was in an AM.ADP state. The rebinding of P_ithen induced detachment from actin leading to a detached M.ADP.P_i state, which produced an alteration in event lifetimes (Figure 4.3a–c).

This scenario suggests that the rebinding of P_i induces detachment by inducing the opening of the cleft in the upper 50 kDa domain, which modulates myosin’s affinity for actin and is putatively coupled to the presence of P_iin the active site (Geeves & Holmes, 2005). Interestingly, this suggests that there might be hysteresis in the transduction process, as myosin appears to be able to bind strongly to actin with P_i still in the active site during force-generation, but rapidly detaches from actin upon P_i -rebinding. We (Debold et al., 2013) and others (Linari, Caremani, & Lombardi,

2010) have suggested a model for this hysteresis based on observations in skeletal muscle myosin II.

In a second test of the P_i-release-first model, we used a myosin Va S1 construct with S217A mutation in the switch I region of the active site, which has been shown to slow P_i-release from the active site by ~10-fold (Forgacs et al., 2009; Gunther et al., 2020; Llinas et al., 2015). Serine 217 (myosin Va numbering) is an absolutely conserved residue that is the first serine in the NDNSSRFG sequence of switch I (Forgacs et al., 2009). Its OH-group putatively forms hydrogen bonds with the gamma-phosphate, ATP and the NH₂ of Arginine 215, and it is thought to participate in proton transfer during the hydrolysis process (Forgacs et al., 2009; Smith & Rayment, 1996). The S217A mutation removes the ability of the OH-group to participate in proton transfer and likely weakens the contacts between switch-I and the gamma-phosphate of ATP (Forgacs et al., 2009).

This mutation also appears to slow the weak- to strong-binding transition, based on observations from solution kinetics; an effect that likely also contributes to the reduced steady-state ATPase rate (Forgacs et al., 2009; Gunther et al., 2020). However, the most relevant effect for the present study is the 10-fold reduction in the release of P_i from actomyosin (Forgacs et al., 2009), which occurs by slowing its release from the active site (Llinas et al., 2015). Importantly, the slowed rate of P_i-release from this construct means it is not reliant on P_i rebinding to the active site as in the experiments with elevated P_i in solution. This provides an independent test of the hypothesis that P_i-release from the active site gates myosin's powerstroke. The predicted outcome in the laser trap assay, however, is the same if P_i-release precedes the powerstroke. Myosin should bind to actin, and only after a long delay that allows for P_i to be released from the active site, should it generate a powerstroke. Contrary to this prediction, we observed that the S217A construct did not affect the size (Figure 4.1e) or rate (Figure 4.2e) of the powerstroke generated, consistent with our

cursory examination of this construct using a less sophisticated analysis (Gunther et al., 2020). Our simulations of a P_i-release-first model confirmed that our analysis of these data had the sensitivity to detect a change in either variable (Figure 4.2b,c). This, like the observation with WT myosin in the presence of P_i, suggests that upon strongly binding to actin, myosin rapidly generates a powerstroke with P_i still in its active site.

These findings and conclusions are consistent with previous FRET-based studies showing that the powerstroke occurs at least twice as fast as P_i-release into solution (Muretta et al., 2015; Trivedi et al., 2015). However, the use of the S217A construct in the present investigation demonstrates that the powerstroke occurs while P_i is still in myosin's active site. Therefore, these findings would be inconsistent with a model in which P_i is released quickly from the active site before stalling the exit tunnel (Houdusse & Sweeney, 2016; Robert-Paganin et al., 2020; Sweeney & Houdusse, 2010) because this mutation prevents entry into the exit tunnel (Llinas et al., 2015). Our conclusion is also consistent with recent findings from cardiac muscle myosin using an ultra-fast laser trap assay where the rate of the powerstroke was estimated to be 1,000. s⁻¹ under a 1.5 pN resistive load (Woody et al., 2019), roughly 50-fold faster than P_i-release for cardiac myosin (Gunther et al., 2019; Malik et al., 2011). Similar rates of the powerstroke were reported in the first description of an ultrafast laser trap assay using fast skeletal muscle myosin II under a 1.5 pN resistive load (Capitanio et al., 2012), a rate 2 orders of magnitude faster than the 20–30. s⁻¹ rate of P_i-release in solution reported for fast skeletal myosin II (Muretta et al., 2015). Interestingly, the rate of the powerstroke has been shown to increase with the applied resistive load (Capitanio et al., 2012; Woody et al., 2019) therefore the rate observed in the present study (500. s⁻¹) under the minimal load of our laser trap assay (7 nm * 0.04 pN/nm = 0.28 pN) appears consistent with the 1000–5000 s⁻¹ rate from 1.5 to 5 pN of resistive load in an ultra-fast laser trap assay using skeletal and cardiac myosin II

(Capitanio et al., 2012; Woody et al., 2019). As indicated above the present findings are difficult to reconcile with the idea that P_i -release precedes the powerstroke, as has been suggested based on high resolution crystal structures (Houdusse & Sweeney, 2016; Robert-Paganin et al., 2020; Sweeney & Houdusse, 2010). Indeed, for our data to be consistent with a P_i -release-first model, P_i would have to be released from the active site at $>1000. s^{-1}$, and then remain in the exit tunnel until it appeared in solution at a rate of $\sim 150. s^{-1}$ for the WT myosin Va construct. P_i would have to remain in the exit tunnel for an even longer duration in the S217A construct, where the rate of P_i -release rate is reduced to $15\text{--}20. s^{-1}$ (Forgacs et al., 2009). Indeed, P_i would need to be released from the active site at $>1000. s^{-1}$, but remain in the exit tunnel for roughly 50 ms, before it is released into solution, at the rate measured for the S217A construct $15\text{--}20. s^{-1}$. This seems an inordinately long duration for P_i to remain trapped in the exit tunnel given that the putative contacts thought to impede P_i 's exit are weak electrostatic interactions. Therefore, a model in which the powerstroke occurs while P_i remains in the active site, and thus precedes P_i release, seems the most plausible explanation for these findings (Figure 4).

Our findings also suggest that myosin transiently exists in a post powerstroke state while P_i remains bound in the active site; a structure that has yet to be captured using x-ray crystallography (Geeves & Holmes, 2005; Llinas et al., 2015; Rayment et al., 1993b; Smith & Rayment, 1996). The reasons for this are unclear, however there are several plausible explanations why this structural state might be difficult to capture; first among these is the inability to crystallize myosin in the presence of actin filaments. This structure is thought to be occupied only while myosin is strongly bound to the actin filament (Geeves & Holmes, 2005; Takagi et al., 2004; Woody et al., 2019), thus it may be extremely difficult to reproduce such a structure without an actin filament. While recent advances in cryo-electron microscopy may present an alternative method to obtain this structure, the speed of both the power-

stroke $500, \text{ s}^{-1}$ (Figure 4.2e) and P_i -release ($150\text{--}200 \text{ s}^{-1}$ [Forgacs et al., 2009; Llinas et al., 2015; Trivedi et al., 2015]) suggest the state is very transient and therefore will be difficult to capture by this methodology. This is supported by the observation that a prepowerstroke-like state can only be captured when myosin is trapped in a transition-like state using ADP and either beryllium, vanadate or aluminum fluoride (Dominguez, Freyzon, Trybus, & Cohen, 1998; Mentes et al., 2018; Reubold, Eschenburg, Becker, Kull, & Manstein, 2003; Smith & Rayment, 1996). It is also clear from single muscle fiber experiments (Debold et al., 2004; Hibberd et al., 1985) and more recently from single molecule studies using an ultra-fast load-clamp laser trap assay (Woody et al., 2019), that myosin's affinity for P_i is strain-dependent, with P_i more readily rebinding to the active site under a high force opposing the direction of the powerstroke, a conclusion also reached based on experiments using myosin Va in a load-clamped laser trap (Sellers & Veigel, 2010). Such a strain would not be present under the conditions currently employed to obtain either x-ray crystallography or cryo-EM structures of myosin.

Our findings are, however, consistent with reports on myosin's structural dynamics in solution using FRET probes, where lever arm rotation is observed to occur very rapidly ($300\text{--}450. \text{ s}^{-1}$) and is at least two-fold faster than the release of P_i into solution (Muretta et al., 2015; Trivedi et al., 2015). Our findings are also consistent with FRET studies demonstrating that, upon strongly binding to actin, the cleft in the actin-binding domain rapidly closes and that this closure is coupled to rotation of the lever arm from the pre- to postpowerstroke state (Conibear et al., 2003; Yengo et al., 2002). And it appears that the closure of the actin-binding cleft occurs prior to the release of P_i , based on a combination of FRET and kinetic experiments (Sun et al., 2008). This is also supported by recent x-ray crystallography structures showing that myosin exhibits a closed cleft during the weak- to strongly bound transition with ADP and P_i still in the active site (Franz et al., 2020). Taken together with the

present observations, these findings suggest that formation of the strong-bond with actin is the structural event that triggers the powerstroke rather than the release of P_i from the active site.

5.3.2 The myosin Va powerstroke occurs in two phases

It is a widely held view that P_i does not readily rebind to myosin in the absence of a significant resistive load or strain (Geeves & Holmes, 2005; Sweeney & Houdusse, 2010; Takagi et al., 2004). However, the present findings demonstrate that in a single molecule laser trap assay, in which myosin experiences very low loads (~ 0.28 pN), elevated levels of P_i altered the distribution of event lifetimes (Figure 4.3b). At the ATP concentration used (100 M), this provides evidence that P_i rebound to the active site in an AM.ADP state, and induced detachment from actin in an AM.ADP. P_i state. Prior experiments using a laser trap assay capable of applying load to an attached cross-bridge suggested that P_i may only rebind to a strained crossbridge using either myosin Va (Sellers & Veigel, 2010) or cardiac myosin (Woody et al., 2019), however lower concentrations of P_i (10 mM) were used in these investigations, therefore we may have observed rebinding at lower loads due to the higher concentration of P_i levels (30 mM) increasing the probability of rebinding. Furthermore, our data suggest that only the longest lasting attachments were eliminated in the presence of elevated P_i (Figure 4.3a), therefore the combination of a high [P_i] and long-lived AM.ADP state may be required to elicit rebinding under minimal resistive strain. Such an effect would be expected to reduce drag on actin filaments under unloaded conditions, as P_i rebinding detaches the longest lived strongly bound crossbridges. Consistent with this idea, elevated P_i levels increase actin filament velocity in an in vitro assay (Debold et al., 2011), and unloaded shortening velocity in skinned single muscle fibers (Pate & Cooke, 1989). P_i -induced acceleration of detachment also appears to occur in myosin

Va, where elevated levels of P_i shorten myosin Va's run length on an actin filament in an unloaded, single molecule, TIRF assay (Baker et al., 2004).

5.3.3 What is the fate of the cross-bridge after P_i rebinds

When phosphate rebinds to myosin II's active site in an AM.ADP state it accelerates detachment from actin (Baker et al., 2002; Debold et al., 2013; Takagi et al., 2004). The P_i -induced decrease in event lifetime observed in the present study (Figure 4.3a) confirms that this also occurs in myosin Va, but what is the fate of the cross-bridge after P_i rebinds to the active site? If the powerstroke can only occur after P_i release from the active site, then it would be expected that the rebinding of P_i would prevent the powerstroke from occurring and then accelerate detachment from actin, as suggested based on structural observations (Llinas et al., 2015; Robert-Paganin et al., 2020; Sweeney & Houdusse, 2010). However, our observation that the powerstroke occurs rapidly upon binding to actin (Figures 1 and 2) is inconsistent with the powerstroke occurring after P_i -release. Our results suggest that myosin's detachment prior to the generation of the powerstroke is not the fate of a cross-bridge following the rebinding of P_i to the active site.

In contrast, if the powerstroke precedes P_i -release from the active site, the powerstroke may be reversed once P_i rebinds to the active site, returning the cross-bridge to the prepowerstroke AM.ADP. P_i state from which detachment occurs rapidly (Dantzig et al., 1992; Takagi et al., 2004). Indeed, in single molecule laser trap assays, displacements in the opposite direction of the original powerstroke have been observed, in particular under a high resistive load, and may occur more readily in the presence of P_i (Takagi et al., 2004; Woody et al., 2019). However, the position of the lever arm was not directly monitored in these prior studies so it is not clear if the backward motion detected was a reverse of the lever arm rotation or if the high resistive load caused myosin to slip on actin (Debold et al., 2005). These observations were

further complicated by the use of low ATP concentrations (1 M), which increases the probability that P_i rebound not to an AM.ADP state but the rigor state, creating an AM. P_i state instead of an AM,ADP. P_i state (Amrute-Nayak et al., 2008).

In the present study we did not see evidence of a reversal of myosin's powerstroke in the presence of elevated P_i , under any condition or using either construct (Figures 1a and 2a). If present, reversals should have been evident as an abrupt reversal of the displacement following a powerstroke while in a low variance state, but this was not seen in the raw displacement records (Figure 4.1a). Even if such reversals occurred with low frequency, they should have been evident as a downward slope at the back end of binding events in ensemble averaging, but again this was not observed for either construct or in the presence of P_i (Figure 4.2a). It is possible that such reversals occurred faster than the time resolution of our instrument (~ 2 ms) and therefore were invisible in the present study. If this was the case it would put time constraints on the rate of the reversal and the duration of the AM.ADP. P_i state following a reversal, indeed both events would have to occur at $>500.$ s^{-1} to have been invisible in our analyses. A recent report, using cardiac myosin II, suggested that reversals occur as slow as $250 s^{-1}$ (Woody et al., 2019), thus it would have to occur more quickly in myosin Va for it to be invisible in the present investigation.

Alternatively, it is possible that myosin detaches from actin without reversing the powerstroke following P_i -rebinding, as has been proposed (Debold et al., 2013). In this type of model, the rebinding of P_i induces detachment from a postpowerstroke state by introducing a P_i -dependent branch in the cross-bridge pathway (blue arrows, Figure 4). This kind of model can also explain the P_i -induced reduction in muscle force and offers an explanation for two observations that have been difficult to reconcile with a model containing a reversal of the powerstroke (Debold et al., 2013). Specifically, the P_i -induced enhancement of actin filament velocity at low pH in the motility assay (Debold et al., 2011; Greenberg & Moore, 2010) and the maintenance of a high

ATPase rate when P_i levels are elevated despite large reductions in isometric force of muscle (Linari et al., 2010). Indeed, in the present study elevated levels of P_i had little or no effect on actin filament velocity in a motility assay using either the WT or S217A construct (Figure 4.3d). If P_i induced reversals of the powerstroke while myosin was strongly bound to actin, we would have expected to see a decrease in Vactin. Thus, a model of the cross-bridge cycle in which the rebinding of P_i can induce detachment from a postpowerstroke state is the most plausible and provides the best explanation of functional and structural findings (Figure 4.4).

In figure 4.4, the pathway followed by wild-type is shown with black arrows and numbers, with the P_i -dependent changes shown with blue arrows. The rates for the S217A are only displayed where they differed from WT myosin (orange numbers). Progressing from left to right, the attachment rate (1 s^{-1}) is taken from the frequency binding events (see Section 2). We did not observe a distinct population of events generating zero displacement (Figure 4.1e) suggesting that the rate of detachment from a prepowerstroke state, if it occurs, is faster than the time resolution of our event detection ($\sim 2\text{ ms}$). The key finding is that for both constructs, and in the absence of presence of 30 mM P_i the powerstroke occurred immediately upon binding to actin, consistent with work from myosin II in an ultra-fast laser trap assay at $1000.\text{ s}^{-1}$ (Capitanio et al., 2012; Woody et al., 2019). No evidence was observed of a reversal of the hitch (dashed line). The P_i -release rate for WT and S217A represent the maximum actin activated rate from (Llinas et al., 2015). Since P_i reduced the bound lifetime (Figure 4.3a) we believe that P_i rebinds to an AM.ADP state and induces detachment from a postpowerstroke state (Debold, Walcott, Woodward, & Turner, 2013). Elevated P_i only eliminated the longest 25% of binding events in the WT myosin and not S217A (Figure 4.3a), thus this pathway was not active in the S217A under the present conditions. We also did not see evidence that P_i reduced the step size, suggesting that myosin in the ADP. P_i state did not readily reattach to actin in

the post powerstroke state after P_i induced detachment (dashed blue line). The ADP-release rate and the secondary powerstroke (i.e., hitch) were combined for simplicity, but may occur in distinct steps (Gunther et al., 2020). Values for ATP-binding and hydrolysis are taken from estimates in solution (Forgacs et al., 2009; Gunther et al., 2020). Hydrolysis and the recovery stroke were combined for simplicity but likely occur at different rates (Gunther et al., 2020).

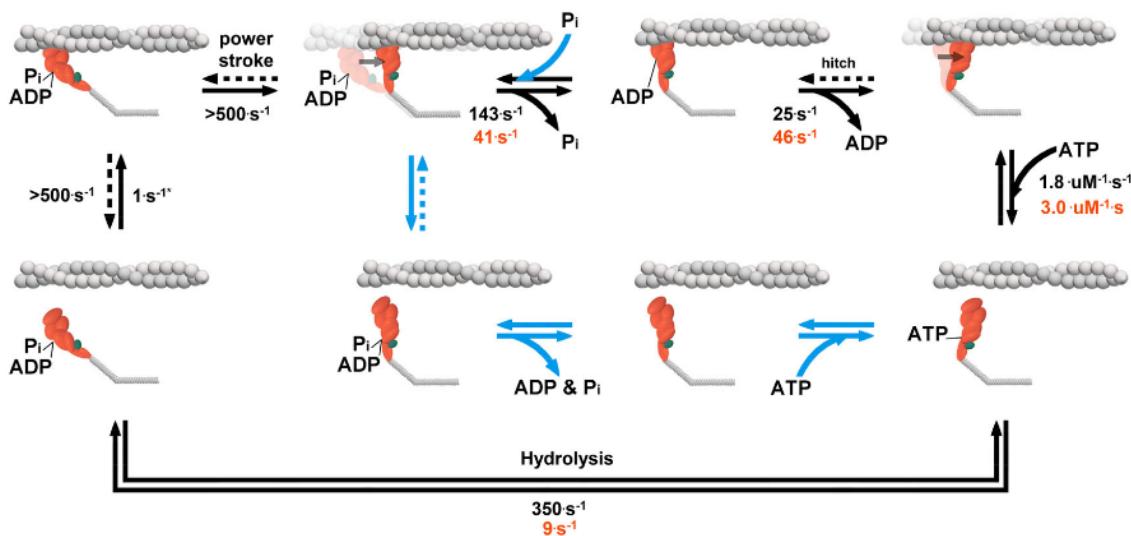


Figure 5.4. Proposed model of cross-bridge cycle. A simplified model postulating the effects of P_i and the S217A myosin's cross-cycle.

5.4 Conclusions

We used two independent approaches to maintain P_i in myosin Va's active site while directly observing its ability to bind to actin and generate a powerstroke. In both cases myosin generated a powerstroke rapidly upon binding to actin, providing strong evidence that myosin generates its powerstroke when P_i is still in its active site. This is consistent with a model in which the formation of the strongly bound state drives rotation of the lever arm and that this key force generating event occurs before the release of P_i from the active site (Muretta et al., 2015; Takagi et al.,

2004; Trivedi et al., 2015). Thus, these findings provide crucial new insight into the molecular mechanism of force generation by myosin, and how it is coupled with the release of the gamma-P_i from myosin, and in doing so reveals new insight into how myosin transduces chemical energy into mechanical work.

5.5 Supplemental Material

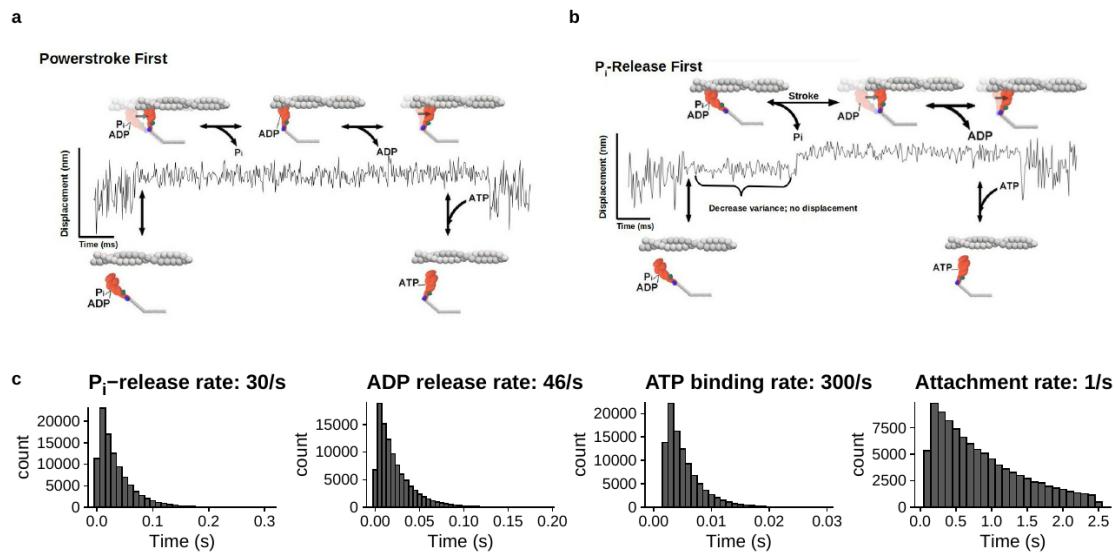


Figure 5.5. Conceptual models of the simulated data.

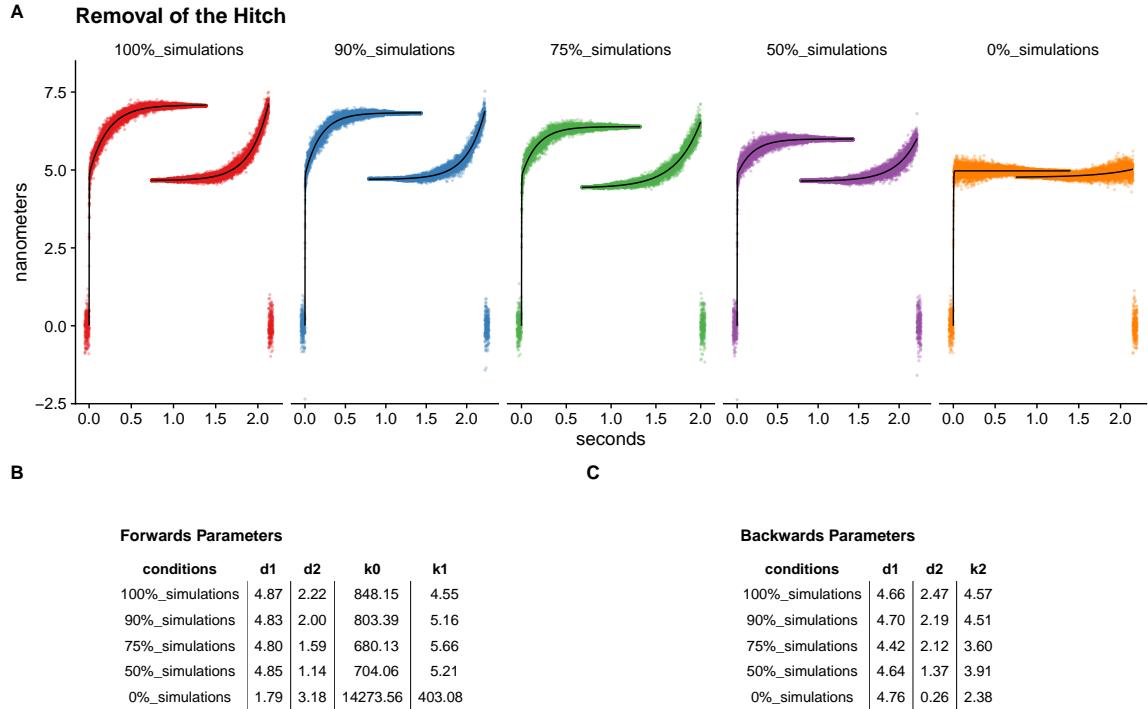
The kinetic scheme for the S217A simulations is based on the values shown in Figure 5.4 of the main text of *Chapter 5*. a) Figures show the two different cross-bridge models used for single molecule simulations. In powerstroke-first simulation (top left) upon binding to actin myosin generated an instantaneous displacement of 6nm, with ADP and P_i still in the active site, 2) P_i is released then from a post-powerstroke AM-D-P_i state, 3) ADP is released and is followed by the completion of the secondary powerstroke or hitch, and 4) myosin is left in a rigor state, AM, until ATP induces detachment at the rate corresponding to the second order dissociation constant and the ATP concentration of 100uM. b) The cross-bridge cycle used for

the P_i -release-first model was the same as the powerstroke-first model except for the timing of the powerstroke. In these simulations upon binding actin myosin myosin does not generate a powerstroke, until P_i was released from a pre-powerstroke AM-D- P_i state at 30s-1. This creates a pause after myosin strongly-binds to the actin filament where there is no net displacement until P_i is released from the active site at which point it generates the same 6nm displacement. 2) ADP is then released from a post powerstroke state, which is followed by a 2nm secondary powerstroke, and 3) the AM rigor state persists until ATP induced detachment at the rate corresponding to the ATP concentration and the second-order binding constant shown in Figure 4 of the main text. c) Bar graphs showing the underlying lifetime distributions used to define the rate of each biochemical transition. Rates for P_i -release, ADP-release, and ATP binding were taken from values reported from solution kinetic studies as indicated in Figure 5.4 of the main text. The attachment rate of 1s^{-1} was taken from the measured frequency in the present data (see Figure 4.3b in the main text). The Brownian motion of the bead-actin-bead assembly was generated by sampling from a Gaussian distribution with a mean of 0 and standard deviation 8nm, the standard deviation was reduced to 4nm during a binding event. These values were chosen to match signal characteristics of the experimentally observed in displacement data records. Each of the above histogram displays representative results from 100,000 random samples from each of the exponential distributions. Distributions were truncated below 1 ms duration. (i.e. 5 datapoints at 5kHz), to match the time resolution of experimental data. The attachment rate was chosen to have a minimum time between events of 100 ms and a maximum time between events of 2.5 seconds, just to limit the size of the datasets generated. A single molecule event was constructed by randomly selecting a value from the power density function of each distribution and this value provided the amount of time that the respective event would spend in each kinetic step. The powerstroke was modeled as instantaneous, occurring in <200μs. In simultaneous

in which the powerstroke occurred prior to P_i -release (powerstroke-first model) the displacement was made to occur simultaneous with the decrease in signal variance. In simulations where the P_i -release occurred prior to the powerstroke the rate of the displacement was the same but only occurred after P_i was release at mean value of 30s⁻¹. All other rates of the kinetic transitions were identical between the two simulation protocols.

5.6 Detecting removal of the hitch

To probe the feasibility of the claim that a rebinding P_i can induce detachment prior to the hitch, which would in turn decrease the overall hitch size as seen in the ensemble averages, we simulated data with various percentages of events that completed the hitch. These simulations serve the purpose to demonstrate that if P_i rebinding does indeed detach myosin before realizing additional displacement from the hitch then we can attempt to estimate what percentage of events would need to be effected by P_i -induced detachment in order to observe a noticeable effect in the ensemble averages. Datasets were simulated with the **{lasertrapr}** app containing 500 single molecule binding events each. All events had a 5nm initial displacement accompanied by a 2nm hitch. The datasets were then analyzed and ensemble averaged using the **{lasertrapr}** app. Forwards ensembles were fit a double exponential and the backwards fit with a single exponential. Full forwards and backwards ensembles are shown in figure 5.6 along with the table of the exponential fit parameters.



In the dataset where all events were allowed to complete the hitch (labeled as “100%_simulations”) the double exponential estimated the initial step (d1) to be 4.8nm and the hitch (d2) 2.2nm. Interestingly, in the datasets that follow where 90%, 75%, and 50% of all events were permitted to proceed with a hitch, the detected hitch was approximately 91%, 73%, and 52% the size as of the control “100%_simulations” group. These simulations show there is a possibility of 1) the ability of the analysis to be sensitive to changes in the size of the hitch, and 2) that the percentage decrease in the measured size of the hitch as estimated from the double exponential fits to the forward ensembles could be used as a method to estimate the percentage of events that did not undergo additional displacement with the hitch. For the case here in regards to high levels of Pi, we could possibly estimate that number/percentage events that were effected/ended by Pi-rebinding. In the real experimental data, the hitch size from Figure 5.2 for the WT 30mM-Pi conditions was 94% of the control conditions indicating that, if these described assumptions were to be true, then only 6% events were effected (i.e. terminated) by Pi-rebinding. This would roughly corre-

spond to 27 events. While the average time on for the WT 0mM-Pi conditions was 230ms, removing the longest 27 events from the experiments data set yields an average attachment time of 132 milliseconds which closely resembles the average attachment time from the WT 30mM-Pi conditions of 145ms.

While analyzing for a decrease in the size of the hitch will most likely provide the most robust method for determining if P_i -induced detachment can reduce the size of the hitch. Another option would be to see if the size of the overall displacement distribution would reduce; however, additional simulation in Figure 5.7 show that this method is less sensitive to the disappearance of the hitch at high ATP concentrations. In these simulations, two data sets were simulated. Both had an initial stroke displacement of 5nm and one had a 2nm and one had no hitches in any events. The data was simulated to mimick higher [ATP] and ATP binding rate (i.e. rigor lifetime) which was simulated with a truncated exponential with a rate of 200/sec (Gunther et al. (2020) measured myoV-WT at 2/uM/sec) and a minimum duration of 5 milliseconds. Data was analyzed with HM-Model/Changepoint wihtin the **{lasertrapr}** app and summarized.

We were unable to detect a difference in the displacement distributions. This is most likely because the time spent at the final displacement (d_2) during “rigor” is too short to influence the total step displacement during the step size calculation. In the single molecule analyzer, the step size is determined by the average of the middle ~90% data points between the changepoint identified start and end of the event. The ends are trimmed to avoid averaging the transitions between populations, but this also includes less data from the final d_2 displacement into the average. Moreover, the ATP concentration would have an effect on this calculation as the calculation of an event’s average displacement essentially is a weighted average of the d_1 and d_2 displacements. The relative weights would then be dependent on the rates of ADP-

release and ATP-binding because these rates would dictate how much time should be spent in a given state/displacement.

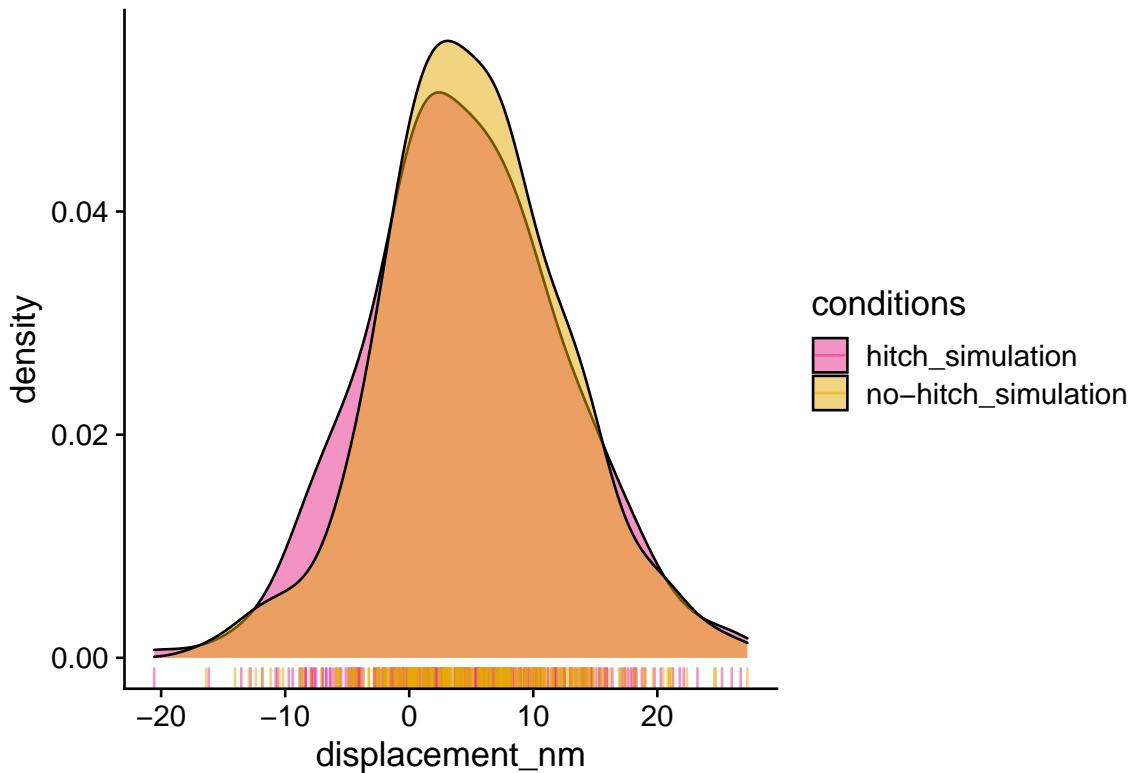


Figure 5.6. Step size comparison. No significant differences found in the average displacement between two simulated conditions with and without the hitch. Plot is interactive online.

```
##  
## Welch Two Sample t-test  
##  
## data: displacement_nm by conditions  
## t = -0.97926, df = 574.57, p-value = 0.3279  
## alternative hypothesis: true difference in means between group hitch_simulation  
## 95 percent confidence interval:  
## -1.854002  0.620343  
## sample estimates:
```

```

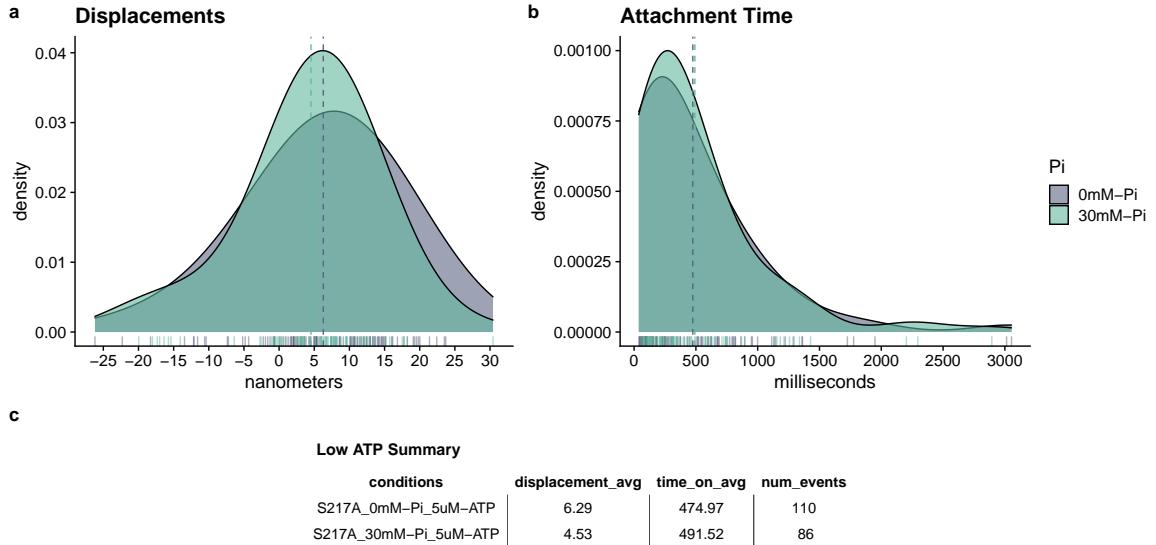
##      mean in group hitch_simulation mean in group no-hitch_simulation
##                                         4.569154                               5.185984

```

5.7 What state is S217A detaching from?

In a previous section we demonstrated that the WT myosin was vulnerable to P_i rebinding as evidenced by a decrease in attachment lifetime and a skewed Q-Q plot. However, for the mutant S217A the attachment lifetimes were unchanged (143 vs 127 ms, 0mM vs 30mM- P_i). With this many questions arose - *Why does the S217A under 0mM- P_i have faster attachment time as compared to the WT?, Why is S217A's event lifetimes largely un-affected by 30mM- P_i ?*. There could be many different ways in which the results could have manifested - *Was the slowed release of P_i from the active site causing a pre-mature detachment?, If P_i is slow to leave the active site, is it also slow to re-bind?, or possibly even Is the S217A impervious to P_i rebinding?*.

In order to test these questions we repeated these experiments with the S217A under the same conditions, but we decreased the ATP concentration to 5uM ATP. If the slowed P_i from the active site was able to cause a premature detachment by “knocking itself off” from actin then decreasing the ATP should not increase the attachment lifetime as it does for WT myosin (Finer, Simmons, and Spudich (1994)). If the attachment time at low ATP concentration is equivalent to the attachment time at high ATP for the S217A, then this result would suggest detachment from a reformed ADP. P_i state. However, if lowering ATP concentration increases the attachment time as compared to the higher ATP conditions, then most likely that S217A would be detaching from a rigor state.



Additional statistical tests show there are no differences between the 0mM-P_i and 30mM-P_i groups at low ATP in regards to displacements:

```
##
## Welch Two Sample t-test
##
## data: low_atp_0$displacement_nm and low_atp_30$displacement_nm
## t = 1.3269, df = 188.46, p-value = 0.1861
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.8518689 4.3528821
## sample estimates:
## mean of x mean of y
## 6.285300 4.534793
```

or for attachment time:

```
##
## Kruskal-Wallis rank sum test
```

```

##  

## data: time_on_ms by pH  

## Kruskal-Wallis chi-squared = 1.0536, df = 1, p-value = 0.3047

```

Since decreasing the ATP concentration increased the attachment lifetime of the S217A, the mutant is most likely detaching from a rigor state, which would also indicate detachment from rigor at the higher ATP concentration in the published results as well. While ATP is most likely ending the events for S217A, the attachment times were still ~two-fold faster in the S217A at high ATP concentration compared to the WT. The simplest explanation is that at high ATP concentration myosin's attachment time is dominated by the ADP-bound state and the mutant does have a ~two-fold faster ADP release rate as measured by solution kinetics (Gunther et al. (2020)).

Perhaps the most interesting finding here is that the S217A mutant was *un-affected* by 30mM-P_i in all experimental conditions (5uM, and 100uM-ATP). There were no changes to displacement or attachment times - *could* that mean that the S217A is not vulnerable to P_i-rebinding? There is no data that supports the ability for the S217A to rebind a P_i in this dataset. However, an important caveat here is that at a combined trap stiffness of 0.04 pN/nm in the present study, multiplied by our 7nm displacements in all conditions, the myosins were experiencing a very low (0.28 pN) resistive load. This is of importance to consider since myosin's ability to rebind P_i is believed to be load sensitive. Which makes you think, *could a mutation in the switch 1 region of myosin's active alter the load dependence of P_i-induced detachment?* This would undoubtedly be a fruitful avenue to explore in future studies.

5.7.1 Exponential fits to attachment times

Since attachment times are exponentially distributed, the data can be transformed from a standard PDF (usually represented by a binned histogram) into a cumulative

density distribution which is interpreted similarly to a survival curve. The empirical cumulative distribution function (ECDF) displays the the percentage of events that have been completed (y-axis) by a certain time period (x-axis). Myosin's attachment time to actin is dependent on many factors which have underlying rates that determine the overall attachment time. These cumulative distributions of attachments times can be fit with 1, 2, or 3 component exponential curves to gain further insight into those rates/processes underlying the attachment times. The form for the triple exponential a la Marco Capitanio et al. (2012) is:

$$y = A1 * (1 - \exp(-k1 * x)) + A2 * (1 - \exp(-k2 * x)) + A3 * (1 - \exp(-k3 * x)) - Alost$$

Removal of the additional exponential components yields the double and single exponential equations. The cumulative distributions for our attachment times are fit with 1, 2, and 3 exponential curves and displayed in Figure 5.8.

Assuming the conclusion from the main text of *Chapter 5* that myosin's power-stroke occurs rapidly upon binding actin and prior to the release of P_i , after force generation the time myosin then spends strongly bound to an actin filament is dependent on three rates 1) P_i -release, 2) ADP-release, and 3) the rate of ATP to rebind and induce detachment. Interestingly, the cumulative distribution of attachment times for the WT under 0mM- P_i was best fit by a triple exponential. Presumably, the rates would reflect the underlying rates of the 3 biochemical states that comprise myosin's attachment time as detailed in the conceptual model in Figure 5.4 in the main text. Moreover, the WT at 30mM- P_i seems more reasonably fit by a 2-exp fit than with no P_i present which supports the model proposed in the main text of a P_i -induced detachment since the addition of P_i removes the need for a third exponential component to best describe the distribution. If this were the case the most likely explanation would be that the rate of the re-binding of P_i would compete with the

rate of ADP-release from the active site, and if P_i rebinds prior to ADP-release then detachment occurs without the need of an ATP-induced detachment eliminating the probability of having longer events and needing the 3rd-exp component to the fit to describe the distribution.

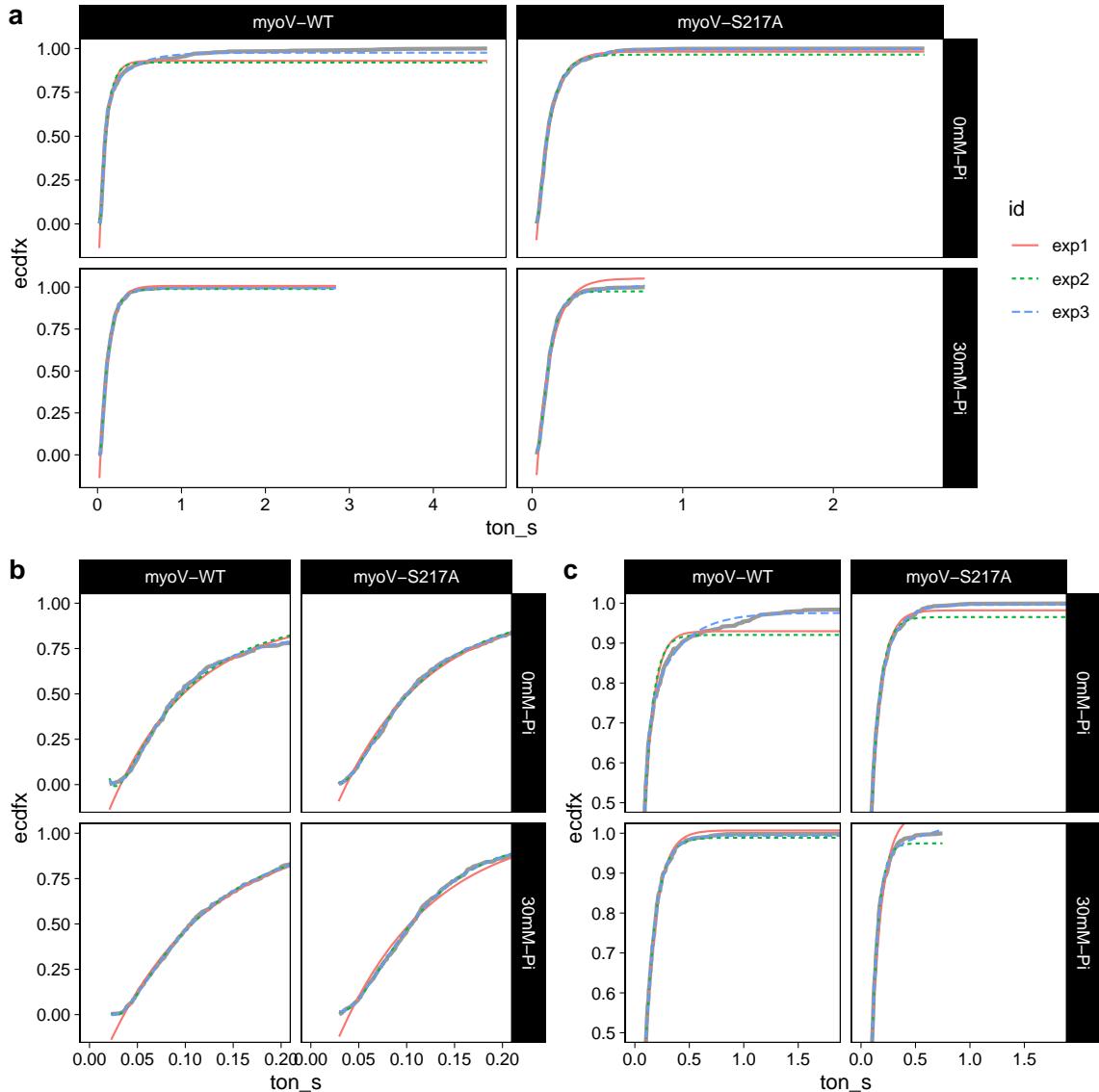


Figure 5.7. Exponential fits to the cumulative distributions of attachment times. The observed data is in the thick grey line. 1-, 2-, and 3-component exponential fits are shown. Plot is interactive online.

Table 5.1. 1-exp parameter estimates

myo	pi	type	Alost	A1	k1
myoV-WT	0mM-Pi	1-exp	0.4350824	1.364900	11.809263
myoV-WT	30mM-Pi	1-exp	0.4177824	1.424976	9.645903
myoV-S217A	0mM-Pi	1-exp	0.4951658	1.477321	10.972758
myoV-S217A	30mM-Pi	1-exp	0.5352932	1.587078	10.220117

Table 5.2. 2-exp parameter estimates

myo	pi	type	Alost	A1	k1	A2	k2
myoV-WT	0mM-Pi	2-exp	-2.8883601	-3.435291	128.75123	1.467507	12.91284
myoV-WT	30mM-Pi	2-exp	-0.8095855	-1.347415	80.56898	1.526716	10.65564
myoV-S217A	0mM-Pi	2-exp	-1.8880768	-2.557349	89.01682	1.634570	12.32603
myoV-S217A	30mM-Pi	2-exp	-0.4275459	-3.709541	30.02066	4.256695	18.20712

Table 5.3. 3-exp parameter estimates

myo	pi	type	Alost	A1	k1	A2	k2	
myoV-WT	0mM-Pi	3-exp	-0.6282920	-5.538877	41.87583	5.5049989	30.53690	0.38
myoV-WT	30mM-Pi	3-exp	-0.4668085	-1.562186	55.33599	0.7026435	32.83311	1.38
myoV-S217A	0mM-Pi	3-exp	-0.5488499	-9.580457	35.12021	9.3694913	30.14432	0.65
myoV-S217A	30mM-Pi	3-exp	-0.4461448	-9.514064	26.36782	9.9907550	21.62430	0.15

CHAPTER 6

ULTRA-COOL CONCLUSION

With the combined efforts of the work in *Chapter 4* and *Chapter 5* which together provide a seemingly robust, precise, and accurate analysis app which was used to analyze single molecule laser trap data under conditions high levels (30mM) of P_i and with our “slow mouse” S217A mutation our primary conclusions reached thus far supports the notion that the powerstroke proceeds P_i -release as we were unable to reject the *null* hypotheses - we saw no effects of the S217A mutation or high levels of P_i (or any interactions) on the absolute displacements or on the rate of unbound-to-bound transitions as revealed via the ensemble averages.

6.1 The “cherry-on-top”

The potential drawbacks of the data from *Chapter 5* is the reliance on data being collected from a mutated myosin and the caveat that the unbound-to-bound transition is not a measurement of the rate of myosin’s actual powerstroke, but is instead the observable rate the reflects the underlying process of myosin’s powerstroke displacing glass beads through solution, a rate heavily damped by viscous drag forces. However, there exists the “better mouse trap” - the ultra-fast force clamp. Advisor and committee members (Debold & Capitanio) have graciously allowed me to include some of their data collected with the UFFC in this dissertation as the dataset is relevant to the main question attempting to be addressed (Powerstroke vs P_i -release first) here. This has allowed me to become more familiar with the UFFC implementation and analysis methods and I think provides a very nice “cherry-on-top” conclusion for the

ending of this work. Moreover, the strength of the data provided by the UFFC offers a strong counter-argument to anyone concerned with any of the potential drawbacks to the concerns mentioned above in regards to *Chapter 5* (not a WT myosin that exists *in-vivo*, indirect measure of transition rate), as UFFC provides a direct quantification/visualization of myosin's powerstroke and affords the time-resolution to use a wild-type myosin that really exists in live cells and organisms.

6.2 Direct observation of myosin's powerstroke

Looking at some of the raw data from the UFFC, several nanometer displacements ($\sim 4\text{nm}$) were observed after the intial binding of myosin to actin. A small portion of raw data is shown in Figure 6.1. The baseline is defined by the wide triangular wave imposed by the QPDs steaming the beads rapidly back and forth to drive the beads at a fixed velocity and force. As the system is designed to keep a constant force on the beads, when an actomyosin interaction occurs causing the feedback to disengage, stopping the oscillation of the beads as the stiffness of the myosin will be exerting the forces required to clamp the force (shown in blue). After a brief dwell period, which is assumed to be a weak binding state, myosin progresses through its powerstroke (green) which is directly observed in the UFFC. Also important to note, is that the *load* is also imposed directly onto the powerstroke as well which is un-obtainable in other trapping assays. This is the only instrument fast enough to apply feedback to the actual *mehanics* of force generation at the level of a single molecule.

The ability for ultrafast to detect these single molecule events with a skeletal muscle myosin II, a very fast low duty ratio motor, at these of high ATP concentrations (1mM) is really just plain *awesome*. For context, Figure 6.2 is a 2-second simulated data trace of what very short 2-10ms data would look like in the standard laser trap setup used in *Chapter 5*.

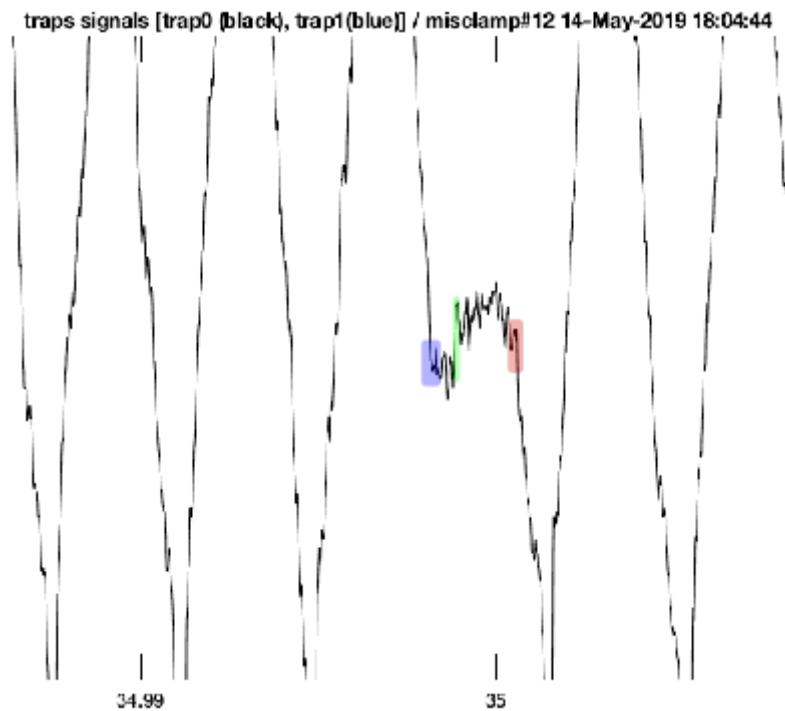


Figure 6.1. Raw data trace from the UFFC showing the initial binding of myosin to actin (blue), direct observation of the powerstroke (green), and ATP induced dissociation (red). Data was collected at 1mM-ATP and 30mM-Pi. Time scale is presumably in seconds so the entirety of the event is ~2ms.

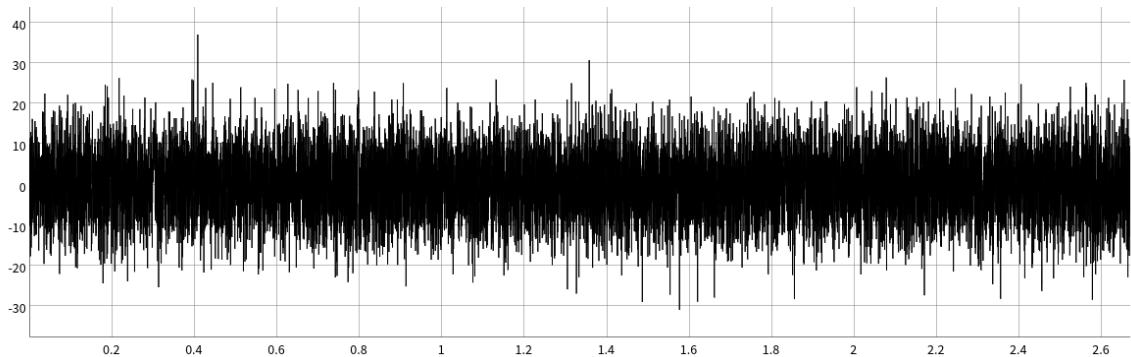


Figure 6.2. There are 5 single molecule displacements in this 2 second simulation that are between 1-10ms in duration. Can you spot them all?

The raw data shown in Figure 6.1 displays a single binding event under high levels of phosphate. This is a similiar response and time-scale to the sequence of binding, dwell, and displacement as observed under no phosphate conditions which indicates that the intial steps in the process of force generation is un-affected by phosphate concentration. This supports the powerstroke first model. The expectation in a P_i -release first would be that the addition of high levels of P_i (and the resistive load applied) would promote P_i to rebind after the initial release which would then *prolong* the initial dwell time prior to the powerstroke. A conceptual model of what a single molecule interaction might look like in the UFFC is shown in Figure 6.3 under a P_i -release first model.

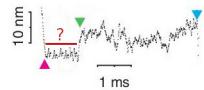


Figure 6.3. A possible conceptual representation of a single UFFC event under a P_i -release first model in high P_i concentration conditions. This effect was not observed.

6.3 Ensemble averages

With a similiar methodology to the ensembles average technique as described in previous chapters, UFFC events can be averaged and ensembled together by their analysis determined start and end of the events. However, an important distinction between the ensemble averages between ultrafast and those shown in previous chapter with a standard laser trap is that the ensembles average shown here more accurately represents the average rate of myosin's powerstroke since the stroke is being directly observed in the single events. For this reason the terminology is carefully chosen when descrbining the intial rates of the two different types of ensembles averages. The wording “unbound-to-bound transition” is used in *Chapter 5* to indicate that we are privy to the notion that this rate is much slower than the true rate of the

powerstroke. Figure 6.4 shows an ensemble averaged data trace from the ultrfast trap collected with fast chicken skeletal (full-length) myosin under high levels of 30mM-Pi. Similarly to the individual events, the ensembles show a rapid displacement after the initial interaction with actin that is un-affected by the addition of P_i into final solution conditions. As is evident in Figure 6.4, even *on average* in the ensembles the powerstroke occurs within several microseconds after the initial actomyosin interaction. The timing of this observation is far greater than any known measured rate of P_i by several orders of magnitude providing strong evidence of the powerstroke preceding P_i -release.

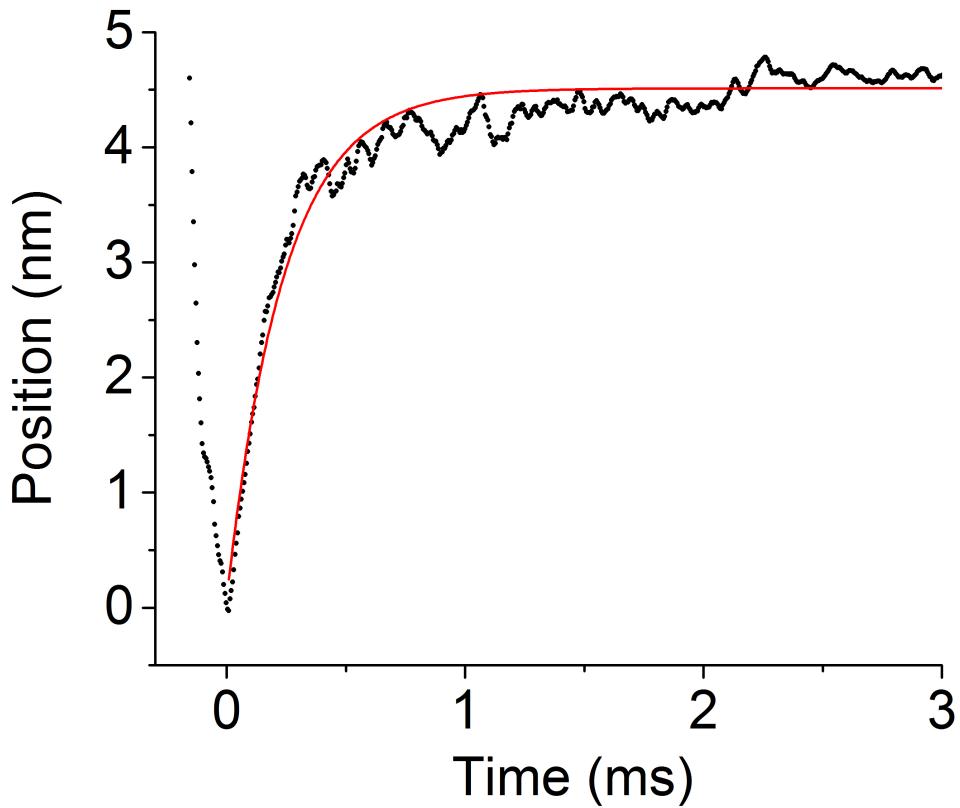


Figure 6.4. Ensemble average of UFFC data at 30mM-Pi. Myosin's powerstroke occurred rapidly within several hundred micro-seconds even under high P_i conditions and substantial resistive load.

CHAPTER 7

THE END

In culmination of the entirety of the work that is relevant to and presented within this dissertation we were unable to provide any evidence in support of a P_i-gated model of molecular force generation in myosin. The data we collected supports the idea that myosin's powerstroke occurs prior to the release of P_i from the active site. While I could go on longer with an extended outro and frame out the importance of this work in the larger context of life, I think there have been plenty opportunities throughout this text where that message has been conveyed (more or less), and as a *kinesiologist* I think going outside now would provide more enjoyable benefits. As such, I think that it is best to be finished now.

Brent

CHAPTER 8

EPILOGUE

This dissertation was written in R/Markdown with the `{bookdown}` package. The PDF output document was styled with formatting adapted from <https://github.com/umasscs/umasstthesis> which provides the latex class files to meet the UMass Amherst grad school styling requirements (note this is an unofficial repository). If you also want to write with R/Markdown you can clone my dissertation repository from <https://github.com/brentscott93/dissy> which will have all of the necessary files/folder setup to run and you can just replace my .Rmd documents with your chapters. Writing with R/Markdown was fun because it allowed me to conditionally render my document to either PDF or HTML output. This was beneficial because it allowed me to display interactive plots when converting to HTML output format so I could host it on my website without needing to keep two separate versions of the paper up to date. Plus, a website link is much easier to share a website link with friends and family.

BIBLIOGRAPHY

- Ashkin, A., J. M. Dziedzic, J. E. Bjorkholm, and Steven Chu. 1986. "Observation of a Single-Beam Gradient Force Optical Trap for Dielectric Particles." *Optics Letters* 11 (5): 288–90. <https://doi.org/10.1364/OL.11.000288>.
- Bailey, Kenneth. 1937. "Composition of the Myosins and Myogen of Skeletal Muscle." *Biochemical Journal* 31 (8): 1406–13.
- Bassett, David R. 2002. "Scientific Contributions of A. V. Hill: Exercise Physiology Pioneer." *Journal of Applied Physiology* 93 (5): 1567–82. <https://doi.org/10.1152/japplphysiol.01246.2001>.
- Blackwell, Thomas, W. Tom Stump, Sarah R. Clippinger, and Michael J. Greenberg. 2021. "Computational Tool for Ensemble Averaging of Single-Molecule Data." *Biophysical Journal* 120 (1): 10–20. <https://doi.org/10.1016/j.bpj.2020.10.047>.
- Block, S M, and K Svoboda. 1995. "Analysis of High Resolution Recordings of Motor Movement." *Biophysical Journal* 68 (4 Suppl): 230S–241S.
- Bloemink, Marieke J., and Michael A. Geeves. 2011. "Shaking the Myosin Family Tree: Biochemical Kinetics Defines Four Types of Myosin Motor." *Seminars in Cell & Developmental Biology, Mechanochemical Cell Biology*, 22 (9): 961–67. <https://doi.org/10.1016/j.semcdb.2011.09.015>.
- Brandt, P W, R N Cox, M Kawai, and T Robinson. 1982. "Effect of Cross-Bridge Kinetics on Apparent Ca²⁺ Sensitivity." *Journal of General Physiology* 79 (6): 997–1016. <https://doi.org/10.1085/jgp.79.6.997>.
- Capitanio, Marco, Monica Canepari, Manuela Maffei, Diego Beneventi, Carina Monico, Francesco Vanzi, Roberto Bottinelli, and Francesco Saverio Pavone. 2012.

- “Ultrafast Force-Clamp Spectroscopy of Single Molecules Reveals Load Dependence of Myosin Working Stroke.” *Nature Methods* 9 (10): 1013–19. <https://doi.org/10.1038/nmeth.2152>.
- Capitanio, M., M. Canepari, P. Cacciafesta, V. Lombardi, R. Cicchi, M. Maffei, F. S. Pavone, and R. Bottinelli. 2006. “Two Independent Mechanical Events in the Interaction Cycle of Skeletal Muscle Myosin with Actin.” *Proceedings of the National Academy of Sciences* 103 (1): 87–92. <https://doi.org/10.1073/pnas.0506830102>.
- Cecchini, Marco, Yuri Alexeev, and Martin Karplus. 2010. “Pi Release from Myosin: A Simulation Analysis of Possible Pathways.” *Structure* 18 (4): 458–70. <https://doi.org/10.1016/j.str.2010.01.014>.
- Chang, Winston, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert, and Barbara Borges. 2021. *Shiny: Web Application Framework for R*. Manual.
- Conibear, Paul B., Clive R. Bagshaw, Piotr G. Fajer, Mihály Kovács, and András Málnási-Csizmadia. 2003. “Myosin Cleft Movement and Its Coupling to Actomyosin Dissociation.” *Nature Structural & Molecular Biology* 10 (10): 831–35. <https://doi.org/10.1038/nsb986>.
- Dantzig, J. A., Y. E. Goldman, N. C. Millar, J. Lacktis, and E. Homsher. 1992. “Reversal of the Cross-Bridge Force-Generating Transition by Photogeneration of Phosphate in Rabbit Psoas Muscle Fibres.” *The Journal of Physiology* 451 (1): 247–78. <https://doi.org/10.1113/jphysiol.1992.sp019163>.
- Dominguez, Roberto, Yelena Freyzon, Kathleen M Trybus, and Carolyn Cohen. 1998. “Crystal Structure of a Vertebrate Smooth Muscle Myosin Motor Domain and Its Complex with the Essential Light Chain: Visualization of the Pre-Power Stroke State.” *Cell* 94 (5): 559–71. [https://doi.org/10.1016/S0092-8674\(00\)81598-6](https://doi.org/10.1016/S0092-8674(00)81598-6).

- DUPUIS, D. E., W. H. GUILFORD, J. WU, and D. M. WARSHAW. 1997. "Actin Filament Mechanics in the Laser Trap." *Journal of Muscle Research & Cell Motility* 18 (1): 17–30. <https://doi.org/10.1023/A:1018672631256>.
- Engelhardt, W. A., and M. N. Ljubimowa. 1939. "Myosine and Adenosinetriphosphatase." *Nature* 144 (3650): 668–69. <https://doi.org/10.1038/144668b0>.
- Fenn, Wallace O. 1924. "The Relation Between the Work Performed and the Energy Liberated in Muscular Contraction." *The Journal of Physiology* 58 (6): 373–95. <https://doi.org/10.1113/jphysiol.1924.sp002141>.
- Finer, Jeffrey T., Robert M. Simmons, and James A. Spudich. 1994. "Single Myosin Molecule Mechanics: Piconewton Forces and Nanometre Steps." *Nature* 368 (6467): 113–19. <https://doi.org/10.1038/368113a0>.
- Forgacs, Eva, Takeshi Sakamoto, Suzanne Cartwright, Betty Belknap, Mihály Kovács, Judit Tóth, Martin R. Webb, James R. Sellers, and Howard D. White. 2009. "Switch 1 Mutation S217a Converts Myosin V into a Low Duty Ratio Motor." *The Journal of Biological Chemistry* 284 (4): 2138–49. <https://doi.org/10.1074/jbc.M805530200>.
- Geeves, M. A., R. S. Goody, and H. Gutfreund. 1984. "Kinetics of Acto-S1 Interaction as a Guide to a Model for the Crossbridge Cycle." *Journal of Muscle Research and Cell Motility* 5 (4): 351–61. <https://doi.org/10.1007/BF00818255>.
- Geeves, M. A., and K. C. Holmes. 1999. "Structural Mechanism of Muscle Contraction." *Annual Review of Biochemistry* 68: 687–728. <https://doi.org/10.1146/annurev.biochem.68.1.687>.
- Gilson, M. K., T. P. Straatsma, J. A. McCammon, D. R. Ripoll, C. H. Faerman, P. H. Axelsen, I. Silman, and J. L. Sussman. 1994. "Open "Back Door" in a Molecular Dynamics Simulation of Acetylcholinesterase." *Science* 263 (5151): 1276–78. <https://doi.org/10.1126/science.8122110>.

- Guilford, W. H., D. E. Dupuis, G. Kennedy, J. Wu, J. B. Patlak, and D. M. Warshaw. 1997. "Smooth Muscle and Skeletal Muscle Myosins Produce Similar Unitary Forces and Displacements in the Laser Trap." *Biophysical Journal* 72 (3): 1006–21. [https://doi.org/10.1016/S0006-3495\(97\)78753-8](https://doi.org/10.1016/S0006-3495(97)78753-8).
- Gulick, A. M., C. B. Bauer, J. B. Thoden, E. Pate, R. G. Yount, and I. Rayment. 2000. "X-Ray Structures of the Dictyostelium Discoideum Myosin Motor Domain with Six Non-Nucleotide Analogs." *The Journal of Biological Chemistry* 275 (1): 398–408. <https://doi.org/10.1074/jbc.275.1.398>.
- Gunther, Laura K., John A. Rohde, Wanjian Tang, Joseph A. Cirilo, Christopher P. Marang, Brent D. Scott, David D. Thomas, Edward P. Debold, and Christopher M. Yengo. 2020. "FRET and Optical Trapping Reveal Mechanisms of Actin Activation of the Power Stroke and Phosphate Release in Myosin V." *Journal of Biological Chemistry* 295 (51): 17383–97. <https://doi.org/10.1074/jbc.RA120.015632>.
- Gunther, Laura K., John A. Rohde, Wanjian Tang, Shane D. Walton, William C. Unrath, Darshan V. Trivedi, Joseph M. Muretta, David D. Thomas, and Christopher M. Yengo. 2019. "Converter Domain Mutations in Myosin Alter Structural Kinetics and Motor Function." *Journal of Biological Chemistry* 294 (5): 1554–67. <https://doi.org/10.1074/jbc.RA118.006128>.
- Harris, D. E., and D. M. Warshaw. 1993. "Smooth and Skeletal Muscle Myosin Both Exhibit Low Duty Cycles at Zero Load in Vitro." *Journal of Biological Chemistry* 268 (20): 14764–68. [https://doi.org/10.1016/S0021-9258\(18\)82398-5](https://doi.org/10.1016/S0021-9258(18)82398-5).
- He, Zhen-He, Rod K. Chillingworth, Martin Brune, John E. T. Corrie, David R. Trentham, Martin R. Webb, and Michael A. Ferenczi. 1997. "ATPase Kinetics on Activation of Rabbit and Frog Permeabilized Isometric Muscle Fibres: A Real Time Phosphate Assay." *The Journal of Physiology* 501 (1): 125–48. <https://doi.org/10.1111/j.1469-7793.1997.125bo.x>.

- Herzog, Walter, Krysta Powers, Kaleena Johnston, and Mike Duvall. 2015. “A New Paradigm for Muscle Contraction.” *Frontiers in Physiology* 6. <https://doi.org/10.3389/fphys.2015.00174>.
- Hill, A. V., and W. Hartree. 1920. “The Four Phases of Heat-Production of Muscle.” *The Journal of Physiology* 54 (1-2): 84–128.
- Hodge, Tony, M. Jamie, and T. V. Cope. 2000. “A Myosin Family Tree.” *Journal of Cell Science* 113 (19): 3353–54. <https://doi.org/10.1242/jcs.113.19.3353>.
- Holmes, Kenneth C., Isabel Angert, F. Jon Kull, Werner Jahn, and Rasmus R. Schröder. 2003. “Electron Cryo-Microscopy Shows How Strong Binding of Myosin to Actin Releases Nucleotide.” *Nature* 425 (6956): 423–27. <https://doi.org/10.1038/nature02005>.
- Houdusse, A., A. G. Szent-Györgyi, and C. Cohen. 2000. “Three Conformational States of Scallop Myosin S1.” *Proceedings of the National Academy of Sciences* 97 (21): 11238–43. <https://doi.org/10.1073/pnas.200376897>.
- Huxley, A. F., and R. Niedergerke. 1954. “Structural Changes in Muscle During Contraction: Interference Microscopy of Living Muscle Fibres.” *Nature* 173 (4412): 971–73. <https://doi.org/10.1038/173971a0>.
- HUXLEY, AF. 1957. “Muscle Structure and Theories of Contraction.” *Prog. Biophys. Biophys. Chem* 7: 255–318.
- Huxley, Hugh, and Jean Hanson. 1954. “Changes in the Cross-Striations of Muscle During Contraction and Stretch and Their Structural Interpretation.” *Nature* 173 (4412): 973–76. <https://doi.org/10.1038/173973a0>.
- Kitamura, Kazuo, Makio Tokunaga, Atsuko Hikikoshi Iwane, and Toshio Yanagida. 1999. “A Single Myosin Head Moves Along an Actin Filament with Regular Steps of 5.3 Nanometres.” *Nature* 397 (6715): 129–34. <https://doi.org/10.1038/16403>.
- Knight, Alex E., Claudia Veigel, Charles Chambers, and Justin E. Molloy. 2001. “Analysis of Single-Molecule Mechanical Recordings: Application to Acto-Myosin

- Interactions.” *Progress in Biophysics and Molecular Biology*, Single Molecule Biochemistry and Molecular Biology, Part II, 77 (1): 45–72. [https://doi.org/10.1016/S0079-6107\(01\)00010-4](https://doi.org/10.1016/S0079-6107(01)00010-4).
- Köhler, Jan, Gerhard Winkler, Imke Schulte, Tim Scholz, William McKenna, Bernhard Brenner, and Theresia Kraft. 2002. “Mutation of the Myosin Converter Domain Alters Cross-Bridge Elasticity.” *Proceedings of the National Academy of Sciences of the United States of America* 99 (6): 3557–62. <https://doi.org/10.1073/pnas.062415899>.
- Kojima, H, E Muto, H Higuchi, and T Yanagida. 1997. “Mechanics of Single Kinesin Molecules Measured by Optical Trapping Nanometry.” *Biophysical Journal* 73 (4): 2012–22.
- König, Karsten. 2000. “Laser Tweezers and Multiphoton Microscopes in Life Sciences.” *Histochemistry and Cell Biology* 114 (2): 79–92. <https://doi.org/10.1007/s004180000179>.
- Llinas, Paola, Tatiana Isabet, Lin Song, Virginie Ropars, Bin Zong, Hannah Benisty, Serena Sirigu, et al. 2015. “How Actin Initiates the Motor Activity of Myosin.” *Developmental Cell* 33 (4): 401–12. <https://doi.org/10.1016/j.devcel.2015.03.025>.
- Longyear, Thomas, Sam Walcott, and Edward P. Debold. 2017. “The Molecular Basis of Thin Filament Activation: From Single Molecule to Muscle.” *Scientific Reports* 7 (1): 1822. <https://doi.org/10.1038/s41598-017-01604-8>.
- Lynn, R. W. 1974. “Actin Activation of Myosin ATPase,” 43th series, 313–28.
- Lynn, R. W., and E. W. Taylor. 1971. “Mechanism of Adenosine Triphosphate Hydrolysis by Actomyosin.” *Biochemistry* 10 (25): 4617–24. <https://doi.org/10.1021/bi00801a004>.
- Margossian, Sarkis S., and Susan Lowey. 1982. “[7] Preparation of Myosin and Its Subfragments from Rabbit Skeletal Muscle.” In *Methods in Enzymology*, 85:55–71.

- Structural and Contractile Proteins Part B: The Contractile Apparatus and the Cytoskeleton. Academic Press. [https://doi.org/10.1016/0076-6879\(82\)85009-X](https://doi.org/10.1016/0076-6879(82)85009-X).
- Mehta, A. D., J. T. Finer, and J. A. Spudich. 1997. "Detection of Single-Molecule Interactions Using Correlated Thermal Diffusion." *Proceedings of the National Academy of Sciences* 94 (15): 7927–31. <https://doi.org/10.1073/pnas.94.15.7927>.
- Mentes, Ahmet, Andrew Huehn, Xueqi Liu, Adam Zwolak, Roberto Dominguez, Henry Shuman, E. Michael Ostap, and Charles V. Sindelar. 2018. "High-Resolution Cryo-EM Structures of Actin-Bound Myosin States Reveal the Mechanism of Myosin Force Sensing." *Proceedings of the National Academy of Sciences* 115 (6): 1292–97.
- Molloy, J E, J E Burns, J C Sparrow, R T Tregear, J Kendrick-Jones, and D C White. 1995a. "Single-Molecule Mechanics of Heavy Meromyosin and S1 Interacting with Rabbit or Drosophila Actins Using Optical Tweezers." *Biophysical Journal* 68 (4 Suppl): 298S–305S.
- Molloy, J. E., J. E. Burns, J. Kendrick-Jones, R. T. Tregear, and D. C. S. White. 1995b. "Movement and Force Produced by a Single Myosin Head." *Nature* 378 (6553): 209–12. <https://doi.org/10.1038/378209a0>.
- Muretta, Joseph M., John A. Rohde, Daniel O. Johnsrud, Sinziana Cornea, and David D. Thomas. 2015. "Direct Real-Time Detection of the Structural and Biochemical Events in the Myosin Power Stroke." *Proceedings of the National Academy of Sciences* 112 (46): 14272–77. <https://doi.org/10.1073/pnas.1514859112>.
- Nesmelov, Yuri E., Roman V. Agafonov, Igor V. Negashov, Sarah E. Blakely, Margaret A. Titus, and David D. Thomas. 2011. "Structural Kinetics of Myosin by Transient Time-Resolved FRET." *Proceedings of the National Academy of Sciences* 108 (5): 1891–96. <https://doi.org/10.1073/pnas.1012320108>.

- Neuman, Keir C., and Attila Nagy. 2008. “Single-Molecule Force Spectroscopy: Optical Tweezers, Magnetic Tweezers and Atomic Force Microscopy.” *Nature Methods* 5 (6): 491–505. <https://doi.org/10.1038/nmeth.1218>.
- PAGE, E. S. 1954. “CONTINUOUS INSPECTION SCHEMES.” *Biometrika* 41 (1-2): 100–115. <https://doi.org/10.1093/biomet/41.1-2.100>.
- Pardee, J. D., and J. A. Spudich. 1982. “Purification of Muscle Actin.” *Methods in Enzymology* 85 Pt B: 164–81. [https://doi.org/10.1016/0076-6879\(82\)85020-9](https://doi.org/10.1016/0076-6879(82)85020-9).
- Patlak, J. 1993. “Measuring Kinetics of Complex Single Ion Channel Data Using Mean-Variance Histograms.” *Biophysical Journal*. [https://doi.org/10.1016/S0006-3495\(93\)81041-5](https://doi.org/10.1016/S0006-3495(93)81041-5).
- R Core Team. 2022. *R: A Language and Environment for Statistical Computing*. Manual. Vienna, Austria: R Foundation for Statistical Computing.
- Rayment, I., W. R. Rypniewski, K. Schmidt-Base, R. Smith, D. R. Tomchick, M. M. Benning, D. A. Winkelmann, G. Wesenberg, and H. M. Holden. 1993. “Three-Dimensional Structure of Myosin Subfragment-1: A Molecular Motor.” *Science* 261 (5117): 50–58. <https://doi.org/10.1126/science.8316857>.
- Reubold, Thomas F., Susanne Eschenburg, Andreas Becker, F. Jon Kull, and Dietmar J. Manstein. 2003. “A Structural Model for Actin-Induced Nucleotide Release in Myosin.” *Nature Structural & Molecular Biology* 10 (10): 826–30. <https://doi.org/10.1038/nsb987>.
- Robert-Paganin, Julien, Olena Pylypenko, Carlos Kikuti, H. Lee Sweeney, and Anne Houdusse. 2020. “Force Generation by Myosin Motors: A Structural Perspective.” *Chemical Reviews* 120 (1): 5–35. <https://doi.org/10.1021/acs.chemrev.9b00264>.
- Scott, Brent, Christopher Marang, Mike Woodward, and Edward P. Debold. 2021. “Myosin’s Powerstroke Occurs Prior to the Release of Phosphate from the Active Site.” *Cytoskeleton* 78 (5): 185–98. <https://doi.org/10.1002/cm.21682>.

- Sellers, James R., and Claudia Veigel. 2010. “Direct Observation of the Myosin-Va Power Stroke and Its Reversal.” *Nature Structural & Molecular Biology* 17 (5): 590–95. <https://doi.org/10.1038/nsmb.1820>.
- Sleep, John, Malcolm Irving, and Kevin Burton. 2005. “The ATP Hydrolysis and Phosphate Release Steps Control the Time Course of Force Development in Rabbit Skeletal Muscle.” *The Journal of Physiology* 563 (3): 671–87. <https://doi.org/10.1113/jphysiol.2004.078873>.
- Smith, C. A., and I. Rayment. 1996. “X-Ray Structure of the Magnesium(II).ADP.vanadate Complex of the Dictyostelium Discoideum Myosin Motor Domain to 1.9 Å Resolution.” *Biochemistry* 35 (17): 5404–17. <https://doi.org/10.1021/bi952633+>.
- Smith, David A., Walter Steffen, Robert M. Simmons, and John Sleep. 2001. “Hidden-Markov Methods for the Analysis of Single-Molecule Actomyosin Displacement Data: The Variance-Hidden-Markov Method.” *Biophysical Journal* 81 (5): 2795–2816. [https://doi.org/10.1016/S0006-3495\(01\)75922-X](https://doi.org/10.1016/S0006-3495(01)75922-X).
- Svoboda, K, and S M Block. 1994. “Biological Applications of Optical Forces.” *Annual Review of Biophysics and Biomolecular Structure* 23 (1): 247–85. <https://doi.org/10.1146/annurev.bb.23.060194.001335>.
- Sweeney, H. Lee, and Anne Houdusse. 2010. “Structural and Functional Insights into the Myosin Motor Mechanism.” *Annual Review of Biophysics* 39 (1): 539–57. <https://doi.org/10.1146/annurev.biophys.050708.133751>.
- Sweeney, H. Lee, Anne Houdusse, and Julien Robert-Paganin. 2020. “Myosin Structures.” In *Myosins*, edited by Lynne M. Coluccio, 1239:7–19. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-38062-5_2.
- Tanaka, Hiroto, Akihiko Ishijima, Makoto Honda, Kiwamu Saito, and Toshio Yanagida. 1998. “Orientation Dependence of Displacements by a Single One-

- Headed Myosin Relative to the Actin Filament.” *Biophysical Journal* 75 (4): 1886–94. [https://doi.org/10.1016/S0006-3495\(98\)77629-5](https://doi.org/10.1016/S0006-3495(98)77629-5).
- “The Nobel Prize in Physiology or Medicine 1922.” n.d. *NobelPrize.org*. <https://www.nobelprize.org/prizes/medicine/1922/summary/>.
- Trivedi, Darshan V., Joseph M. Muretta, Anja M. Swenson, Jonathon P. Davis, David D. Thomas, and Christopher M. Yengo. 2015. “Direct Measurements of the Coordination of Lever Arm Swing and the Catalytic Cycle in Myosin V.” *Proceedings of the National Academy of Sciences* 112 (47): 14593–98. <https://doi.org/10.1073/pnas.1517566112>.
- Trivedi, Darshan V., Joseph M. Muretta, Anja M. Swenson, David D. Thomas, and Christopher M. Yengo. 2013. “Magnesium Impacts Myosin V Motor Activity by Altering Key Conformational Changes in the Mechanochemical Cycle.” *Biochemistry* 52 (27): 4710–22. <https://doi.org/10.1021/bi4004364>.
- Trivedi, Darshan V., Suman Nag, Annamma Spudich, Kathleen M. Ruppel, and James A. Spudich. 2020. “The Myosin Family of Mechanoenzymes: From Mechanisms to Therapeutic Approaches.” *Annual Review of Biochemistry* 89 (1): 667–93. <https://doi.org/10.1146/annurev-biochem-011520-105234>.
- Tyska, M. J., D. E. Dupuis, W. H. Guilford, J. B. Patlak, G. S. Waller, K. M. Trybus, D. M. Warshaw, and S. Lowey. 1999. “Two Heads of Myosin Are Better Than One for Generating Force and Motion.” *Proceedings of the National Academy of Sciences* 96 (8): 4402–7. <https://doi.org/10.1073/pnas.96.8.4402>.
- Tyska, Matthew J., and David M. Warshaw. 2002. “The Myosin Power Stroke.” *Cell Motility* 51 (1): 1–15. <https://doi.org/10.1002/cm.10014>.
- Unger, Matthew, and Edward P. Debold. 2019. “Acidosis Decreases the Ca²⁺ Sensitivity of Thin Filaments by Preventing the First Actomyosin Interaction.” *American Journal of Physiology-Cell Physiology* 317 (4): C714–18. <https://doi.org/10.1152/ajpcell.00196.2019>.

- Veigel, Claudia, Marc L. Bartoo, David C. S. White, John C. Sparrow, and Justin E. Molloy. 1998. "The Stiffness of Rabbit Skeletal Actomyosin Cross-Bridges Determined with an Optical Tweezers Transducer." *Biophysical Journal* 75 (3): 1424–38. [https://doi.org/10.1016/S0006-3495\(98\)74061-5](https://doi.org/10.1016/S0006-3495(98)74061-5).
- Veigel, Claudia, Lynne M. Coluccio, James D. Jontes, John C. Sparrow, Ronald A. Milligan, and Justin E. Molloy. 1999. "The Motor Protein Myosin-I Produces Its Working Stroke in Two Steps." *Nature* 398 (6727): 530–33. <https://doi.org/10.1038/19104>.
- Veigel, Claudia, Justin E. Molloy, Stephan Schmitz, and John Kendrick-Jones. 2003. "Load-Dependent Kinetics of Force Production by Smooth Muscle Myosin Measured with Optical Tweezers." *Nature Cell Biology* 5 (11): 980–86. <https://doi.org/10.1038/ncb1060>.
- Veigel, Claudia, Fei Wang, Marc L. Bartoo, James R. Sellers, and Justin E. Molloy. 2002. "The Gated Gait of the Processive Molecular Motor, Myosin V." *Nature Cell Biology* 4 (1): 59–65. <https://doi.org/10.1038/ncb732>.
- Walker, Matthew L., Stan A. Burgess, James R. Sellers, Fei Wang, John A. Hammer, John Trinick, and Peter J. Knight. 2000. "Two-Headed Binding of a Processive Myosin to F-actin." *Nature* 405 (6788): 804–7. <https://doi.org/10.1038/35015592>.
- Warshaw, David M., William H. Guilford, Yelena Freyzon, Elena Krementsova, Kimberly A. Palmiter, Mathew J. Tyska, Josh E. Baker, and Kathleen M. Trybus. 2000. "The Light Chain Binding Domain of Expressed Smooth Muscle Heavy Meromyosin Acts as a Mechanical Lever*." *Journal of Biological Chemistry* 275 (47): 37167–72. <https://doi.org/10.1074/jbc.M006438200>.
- Woodward, Mike, Eric Ostrander, Seung P. Jeong, Xiarong Liu, Brent Scott, Matt Unger, Jianhan Chen, Dhandapani Venkataraman, and Edward P. Debold. 2020. "Positional Isomers of a Non-Nucleoside Substrate Differentially Affect Myosin

Function.” *Biophysical Journal* 119 (3): 567–80. <https://doi.org/10.1016/j.bpj.2020.06.024>.

Woody, Michael S, Donald A Winkelmann, Marco Capitanio, E Michael Ostap, and Yale E Goldman. 2019. “Single Molecule Mechanics Resolves the Earliest Events in Force Generation by Cardiac Myosin.” Edited by Andrew P Carter, Olga Boudker, Michael Geeves, and Arne Gennerich. *eLife* 8 (September): e49266. <https://doi.org/10.7554/eLife.49266>.

Yount, R G, D Lawson, and I Rayment. 1995. “Is Myosin a ”Back Door” Enzyme?” *Biophysical Journal* 68 (4 Suppl): 44S–49S.