Obscurin is a semi-flexible molecule in solution

Jacob Whitley

Follow this and additional works at: https://commons.lib.jmu.edu/honors201019

Part of the Other Chemistry Commons

Recommended Citation
Whitley, Jacob, "Obscurin is a semi-flexible molecule in solution" (2019). Senior Honors Projects, 2010-current. 729.
https://commons.lib.jmu.edu/honors201019/729

This Thesis is brought to you for free and open access by the Honors College at JMU Scholarly Commons. It has been accepted for inclusion in Senior Honors Projects, 2010-current by an authorized administrator of JMU Scholarly Commons. For more information, please contact dc_admin@jmu.edu.
Obscurin is a semi-flexible molecule in solution

An Honors College Project Presented to

the Faculty of the Undergraduate

College of Science and Mathematics

James Madison University

by Jacob Allen Whitley

May 2019

Accepted by the faculty of the Department of Chemistry and Biochemistry, James Madison University, in partial fulfillment of the requirements for the Honors College.

FACULTY COMMITTEE: HONORS COLLEGE APPROVAL:

Project Advisor: Nathan T. Wright, Ph.D. Bradley R. Newcomer, Ph.D.,
Associate Professor, Department of Chemistry and Dean, Honors College
Biochemistry

Reader: Gina Macdonald, Ph.D. Reader: Isaiah C. Sumner, Ph.D.
Professor, Department of Chemistry and Associate Professor, Department of Chemistry and
Biochemistry Biochemistry

Reader: ,

PUBLIC PRESENTATION
This work is accepted for presentation, in part or in full, at the 63rd annual biophysical society meeting on March 2-6 2019.
Table of Contents

List of Figures 4

Acknowledgements 6

Abstract 7

Chapter 1 – Project Background 8
  Background 8
  Nuclear Magnetic Resonance (NMR) 11
  Small Angle X-ray Scattering (SAXS) 13
  Molecular Dynamics/Steered Molecular Dynamics (MD/SMD) 14

Chapter 2 – Experimental Results 17
  Introduction 17
  Results 19
  Discussion 37
  Conclusions 42
  Materials and Methods 42

Appendix 47

References 65
List of Figures

Chapter 1

Figure 1-1. Trainlike Structure of Obscurin ........................................ 8
Figure 1-2. Architecture of Obscurin ................................................. 10
Figure 1-3. Cartoon Diagram of NMR Spectroscopy ....................... 12
Figure 1-4. SAXS Data Collection .................................................. 14
Figure 1-5. Diagram of Muscle Cell .................................................. 15

Chapter 2

Table 2-1. Linker Composition Between Human Obscurin Ig Domains 20
Table 2-2. PDB Accession Numbers of Solved Human Obscurin Ig Domains 22
Figure 2-1. Solution Structure of Ig57 ........................................... 23
Table 2-3. NMR-Derived Restraints and Statistics of 20 NMR Structures of Ig57 24
Figure 2-2. NMR Examples of Ig57 Experiments .......................... 25
Figure 2-3. NMR Dual Domain Construct Analysis ....................... 26
Figure 2-4. SAXS Analysis of a Representative Obscurin Dual Domain Construct 27
Figure 2-5. Guinier and RDC Analysis ........................................... 28
Figure 2-6. Guinier Plot and Guinier Region Zoom In for Each Dual Domain System 28
Figure 2-7. Comparison of MultiFoXS Fits to Experimental SAXS Data 29
Figure 2-8. MD Analysis of Dual Domain Constructs .................... 31
Figure 2-9. 800 nanosecond MD Analysis of Ig58/59 ....................... 32
Figure 2-10. Representative SMD Analysis .................................... 34
Figure 2-11. SMD Simulations for Each Dual Domain System 35

Figure 2-12. Simplified Model of Obscurin Dynamics in Solution 39

Figure 2-13. Modeled Average Length of Obscurin 40
Acknowledgements

I would like to acknowledge Dr. Nathan T. Wright for allowing me to work with him in his research laboratory. I would also like to acknowledge Aidan M. Ex-Willey, Daniel R. Marzolf, Dr. Maegen A. Ackermann, Dr. Anthony L. Tongen, and Dr. Oleksandr Kokhan for their substantial collaboration on this project. Additionally, the entire JMU Department of Chemistry and Biochemistry has been instrumental in my progress with this research. This work was supported by Research Corporation Cottrell College Award 22450, NSF REU award (CHE-1062629), and NSF RUI (MCB-1607024) (Nathan T. Wright) and grants from National Institutes of Health (HL116778) (Maegen A. Ackermann). SAXS data were collected using resources of the Advanced Photon Source, a U.S. Department of Energy (DOE) Office of Science User Facility operated for the DOE Office of Science by Argonne National Laboratory under Contract No. DE-AC02-06CH11357. This work was presented at the 63rd annual Biophysical Society meeting on March 2-6, 2019.
Abstract:

Obscurin, a giant modular cytoskeletal protein, is comprised mostly of tandem immunoglobulin-like (Ig-like) domains. This architecture allows obscurin to connect distal targets within the cell. The linkers connecting the Ig domains are usually short (3-4 residues). The physical effect arising from these short linkers is not known; such linkers may lead to a stiff elongated molecule or, conversely, may lead to a more compact and dynamic structure. In an effort to better understand how linkers affect obscurin flexibility, and to better understand the physical underpinnings of this flexibility, here we study the structure and dynamics of four representative sets of dual obscurin Ig domains using experimental and computational techniques. We find in all cases tested, tandem obscurin Ig domains interact at the poles of each domain and tend to stay relatively extended in solution. NMR, SAXS and MD simulations reveal that while tandem domains are elongated, they also bend and flex significantly. By applying this behavior to a simplified model, it becomes apparent obscurin can link targets more than 200 nm away. However, as targets get further apart, obscurin begins acting as a spring, and requires progressively more energy to further elongate.
Chapter 1

Background:

Obscurin is an extremely large (~970 kDa) human protein. Proteins are biological macromolecules that perform tasks inside of cells: think of them as biological nanobots. Proteins are long linear polymers of amino acids, which fold and contort into unique shapes that dictate their function. Obscurin is a cytoskeletal protein, which means it functions as scaffolding or structural support in cells. Obscurin is found in both muscle cells and epithelial (skin, breast tissue, intestinal linings, etc.) cells of humans, and is made up of multiple domains (parts of proteins that are able to function normally when isolated from the rest of the protein) linked together. This molecular structure is similar to that of a long train (Figure 1-1): each domain is a train car, and the linker between the cars is slightly flexible.

The purpose of the work described in this thesis is to determine how obscurin behaves in the cell by analyzing its flexibility. Due to its structure and function, we expect obscurin to also behave as a force resistor in cells. Other known cytoskeletal force resistors are similar in structure to obscurin, and are known to resist force.

Figure 1-1: A train representing the structure of obscurin. The train cars are similar to the domains which are linked together in a long chain.
through stretch. The flexibility and behavior of obscurin when stretched is not well studied. Obscurin studies will provide insights into how our cells use the cytoskeleton to translate physical force into biochemical signals and resist strain. Our approach is to examine obscurin very closely outside the cellular milieu. This provides us with a simplified model, which we can then extrapolate into the entire cell. Likewise, since obscurin is so large, we cannot study the entire protein but instead small parts of the protein that, for a variety of reasons, are likely to be representative of how the rest of the molecule works. In this way, this work is highly reductive. However, due to the technical limits of the techniques that we employ, this simplified version of obscurin must be used.

In this introductory chapter, I explain why we study obscurin, why we chose to study the regions we chose, and how the biochemical techniques we use can provide visual details and functional clarity to a world that is far too small for even the most powerful microscope to visualize.

While obscurin is small relative to everyday objects, it is extremely large relative to many other proteins. The size and nature of obscurin creates many difficulties in isolating and studying it, due to the fact that traditional protein structure analysis techniques are not applicable to the whole obscurin protein. NMR is only suitable for proteins less than 30 kDa, X-ray crystallography does not work well with flexible systems, and Cryo-EM requires pure soluble protein (which thus far precludes studying obscurin). Thus, we use a reductive approach in studying its flexibility.
The architecture of obscurin is similar to a long train (Figure 1-1). Each train car represents a single domain, linked together in a long chain. There are different types of domains in obscurin, but the most common is the immuno-globulin like (Ig) domain. There are approximately 60 of these Ig domains in the full protein, and they are mostly connected by short linkers (4-5 amino acids between the domains) (Figure 1-2). There are two types of short linkers, those that contain proline (an amino acid) and those that do not. Prior knowledge of proline in protein structures suggests short proline linkers would be less flexible than short linkers without proline. We chose four sets of dual domain systems: two with short proline linkers and two short linkers without proline. These dual domain systems are representative of ~70% of the obscurin molecule (Figure 1-2).

Obscurin is known to have both structural and functional roles in cells, and thus it appears likely that the protein is somehow involved in force resistance. Here, we attempt to more directly test obscurin’s ability to bend and flex. We used dual domain systems (two domains and the linker connecting them) to explain the behavior of the entire protein. Since the domains themselves are functional apart from the rest of obscurin, it is clear that the results gathered from these dual domain systems can be extrapolated for the whole obscurin protein. Most of the domains we studied here have previously solved
structures using nuclear magnetic resonance (NMR). These domains were not entirely representative of the global obscurin protein (two sets of domains have proline linkers and one set has non-proline linkers. In order to create a more representative study, the NMR structure of another domain (to form another set of domains with a non-proline linker) is necessary.

**Nuclear Magnetic Resonance (NMR)**

Our lab specializes in protein NMR structure, function, and analysis studies. NMR is an experimental technique that can be performed on molecules and proteins, which can show the chemical environment of each atom or amino acid residue in the protein. When atoms are in different chemical environments, they have different NMR signals that elucidate structural information. In order to solve a protein structure using NMR, there are a number of experiments that must be performed. The protein must first be labelled using $^{15}$N and $^{13}$C, isotopes of nitrogen and carbon that can be recognized by the NMR. For small molecules, one-dimensional NMR experiments are sufficient. Two and three dimensional NMR experiments are necessary to completely solve the much larger protein structure. These dimensions are analogous to a book. One-dimensional experiments are similar to a word or a line of a page, two-dimensional experiments are similar to an entire page of text, and three-dimensional experiments are similar to the thickness of the book. When the protein sample is placed into the NMR, it is subjected to an external
This magnetic field affects each atom of the sample differently depending on where they are located in the structure. One experiment, heteronuclear single quantum coherence (HSQC), is two dimensional and is similar to a roadmap for proteins. Each amino acid in the protein has its own peak on the spectrum. This experiment is used in conjunction with the three-dimensional experiments to determine which peaks are associated with each amino acid.

In addition to three dimensional experiments, nuclear Overhauser effects (NOEs) are used to determine a protein structure. NOESYs are experiments that show which atoms are near each other in the three-dimensional space of the protein structure. This helps to create an idea of the structure of the protein, because NOEs show which amino acids are near each other when the protein is fully folded. These experiments allow us
to assign each amino acid to certain peaks on the NMR spectra, and create an accurate and high resolution structure (meaning we are highly confident in the positions of each atom in the structure within a small range) of the protein of interest. Simulated annealing programs are also used in order to determine the structure of the protein. These computer programs use the NMR data and assignments to simulate folding the protein into different conformations, giving violations (if the conformations do not match the NMR data) and energy values (the average energy value of each atom in the structure calculated from a forcefield similar to what is described in the MD section). A structure is considered ‘good’ when the programs give very few violations, and all of the conformations agree with one another.

NMR chemical shift perturbation maps show differences between protein structures. Differences occurring between the spectra indicate the areas of the proteins interact. When proteins interact, the chemical environment around their interaction site is changed, which produces a change in the NMR spectrum. Peaks on the spectra that overlap indicate that the areas of the proteins to not significantly interact with each other.

**Small Angle X-ray Scattering (SAXS)**

SAXS is another experimental technique used to determine the conformation of the dual domains in solution. The protein sample is illuminated by an X-ray beam, the beam is scattered off of the protein in solution, and this creates a scattered intensity profile (Figure 1-4). When analyzed, the intensity and scattering of the pattern provide information
about the conformation/orientation of the protein. SAXS data is obtained in reciprocal space, so a Fourier transform must be performed in order to analyze the data in real space. This transformation along with a Guinier analysis (a mathematical technique to analyze SAXS data) provides an $R_g$ value, or a radius of gyration. Since the protein sample is in solution, this protein exists in many conformations. The $R_g$ value is an average of all of the conformations of that sample, as it can freely move in solution. The $R_g$ value is used to find the length of the molecule, and this information helps to elucidate the average conformation of molecules in solution.

**Molecular Dynamics/Steered Molecular Dynamics (MD/SMD)**

In addition to the experimental techniques above, we also performed MD and SMD simulations on the representative constructs. These computational techniques allowed us to analyze the domains more closely (on an atom by atom basis). MD is a simulation, or computer model. A protein structure model is placed in an environment full of water molecules and the simulation predicts how the protein will behave. The program treats
each chemical bond as a classical spring and uses Newton’s laws of motion to predict where each atom would be after a small amount of time (2 x 10^{-15} seconds). After this initial time step, the program recalculates the location of each atom, and this process continues until a sufficient amount of time has been reached (generally in the 10^{-5}/10^{-6} timescale). A forcefield accounts for various forces that molecules would experience, such as bond stretching, bond bending, bond twisting, and non-bonding terms such as electrostatic interactions. SMD is similar to MD, but instead of the protein model moving randomly, SMD samples low-probability states that would be unlikely to occur in MD. In this case, we simulated what would happen to the dual domains if they were pulled apart at a constant rate (0.1 m/s). This pulling adds another spring to the system, attached to both ends of the dual domains. While unlikely to occur in MD, this SMD simulation represents the type of movement and stretch that obscurin would be subjected to in

![Figure 1-5: Diagram of a muscle cell. Obscurin is shown in blue, connecting the sarcoplasmic reticulum to the contractile apparatus of the muscle. The sarcoplasmic reticulum can move, and obscurin tethers it.](image-url)
muscle cells (Figure 1-5). It is easy to imagine how obscurin (blue) would stretch if the gray areas began to pull apart from each other.

Chapter 1 is a brief overview and explanation of the techniques described in chapter 2. The next chapter will explain the experimental results in more detail.
Chapter 2

Introduction

Most cells in the body are subjected to motion, ranging from muscle cells contracting and relaxing to epithelial cells conforming to body movement. Yet cells also must be physically strong to maintain homeostasis and normal architecture amidst this strain. Giant cytoskeletal proteins are long, chain-like molecules that connect distal cellular regions and have the capacity to bend and stretch. Thus, these proteins provide a potential mechanism to assist the cell in its capability to be both flexible and strong.

The most well known giant cytoskeletal protein is titin. This protein spans from the Z-disk to the M-band in myocytes, and is mostly comprised of hundreds of consecutive, individually-folded Ig domains. Through a domain unraveling mechanism, titin acts as a molecular spring, resisting stretch force longitudinally as the muscle cell overextends.

Obscurin, another giant cytoskeletal protein, has a similar architecture to titin. This protein can be found in at least 20 different forms, ranging from 20 kDa to 970 kDa. At its longest, the N-terminal two-thirds of the protein is comprised of over 60 tandem Ig and Fibronectin (FnIII)-like domains connected to their neighbors via short linkers. The C-terminus contains multiple signaling domains (i.e. PH, RhoGEF, IQ) and either an ankyrin binding region (in obscurin A isoforms) or kinase domains (in obscurin B isoforms). Obscurin’s multiple functions are closely linked to its complex subcellular localization. In skeletal muscles, the ankyrin binding region of obscurin A
binds to small ankyrin 1 (sAnk1) at the sarcoplasmic reticulum.\textsuperscript{20,24–26} Ablation of this interaction reduces sAnk1 levels, which in turn leads to aberrant Ca\textsuperscript{2+} homeostasis.\textsuperscript{27–29} Likewise, obscurin interacts with ankyrin-B in the costamere. When this interaction is disrupted, skeletal muscles experience increased exercise-induced damage due to the improper assembly of the dystrophin complex.\textsuperscript{30} Obscurin B binds to and phosphorylates N-cadherin at the intercalated disk in cardiomyocytes, suggesting that it may modulate muscle cell adhesion.\textsuperscript{21} Complementing these membrane-associated interactions, obscurin binds to the sarcomeric contractile apparatus in several locations.\textsuperscript{5} The 58\textsuperscript{th}-59\textsuperscript{th} obscurin Ig-like domains form a complex with the titin ZIg9 domain at the Z-disk during development, suggesting obscurin plays a role in myofibrillogenesis.\textsuperscript{18,28,31–33} Additionally, the N-terminus of obscurin interacts with titin, slow myosin binding protein C, and myomesin at the M-band, contributing to the M-band lattice assembly, structure, and strength.\textsuperscript{34–37} Thus, obscurin forms the only known connection between the muscle contractile apparatus and the surrounding membrane structures.\textsuperscript{5,34,38,39} Clinically, obscurin is linked to breast and colorectal cancers, and obscurin knockdown cells undergo epithelial-to-mesenchymal transition.\textsuperscript{40–42} In muscle, specific obscurin mutations that alter target protein binding are causally linked to hypertrophic cardiomyopathy, dilated and restricted cardiomyopathy, and muscular dystrophy.\textsuperscript{38,43–47}

In order to better understand both how obscurin exists in solution and responds to stretch, here we study a series of representative tandem obscurin Ig domains using structural biology and computation techniques. We find these dual domain constructs are predominantly extended in solution, yet the domains are also moderately mobile relative
to each other. This finding led to the question of how these domains could be extended (suggesting a framework to maintain this conformation) and also flexible (suggesting that there is not a significant framework present). MD simulations suggest that transient noncovalent bonds between mobile regions in neighboring domains are largely responsible for these dual domains being extended yet dynamic.

Results

Implicit in the observation that obscurin links various cellular targets to each other is the fact that the protein must act as a tether. While obscurin-target interactions in muscle are increasingly well documented, the conformation and dynamics of the obscurin region between these anchor points (the tether) are less understood. Here, we investigate obscurin’s conformation in solution. In an effort to more easily collect high-resolution information about this protein, we utilized a reductive approach and studied a series of representative obscurin dual-domain systems. The linkers between obscurin domains can be broadly divided into short linkers containing proline residues (48% of all obscurin linkers), short linkers with no proline residues (22%), and long linkers (>6 residues) (30%) (Table 2-1). Previous studies, plus basic biochemistry knowledge, suggest the proline-containing linkers may be more rigid, and long linkers are almost certainly more flexible. Here we study two constructs with proline-containing linkers and two constructs with proline-absent linkers to better understand the mobility these short linkers confer on the obscurin molecule as a whole.
Table 2-1:
The linker between each human obscurin Ig domain
Numbered Ig domains from obscurin (CAC44768.1)

Linkers: We define the linker as the residue following the last lysine (or arginine) in one domain to the residue preceding the first lysine (or arginine) in the next domain. These were all aligned to the Ig34-39 region, and include the final and first residues of the flanking domains.

Note that in this nomenclature, Ig57, Ig58, and Ig59 are numbered 47, 48, and 49.

<table>
<thead>
<tr>
<th>Type of Linker</th>
<th>Linked Domains</th>
<th>Linker Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long</td>
<td>1-2</td>
<td>daeaacaeqaph</td>
</tr>
<tr>
<td>Long</td>
<td>2-3</td>
<td>dsdaadtasrpgtstaallahlqrrreamraegapasp</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>3-4</td>
<td>repavp</td>
</tr>
<tr>
<td>Short non-proline containing</td>
<td>4-5</td>
<td>qgnl</td>
</tr>
<tr>
<td>Long</td>
<td>5-6</td>
<td>saprkppl</td>
</tr>
<tr>
<td>Long</td>
<td>6-7</td>
<td>vhlapkla</td>
</tr>
<tr>
<td>Long</td>
<td>7-8</td>
<td>raapgltankp</td>
</tr>
<tr>
<td>Long</td>
<td>8-9</td>
<td>qglarflh</td>
</tr>
<tr>
<td>Long</td>
<td>9-10</td>
<td>sepkvvfak</td>
</tr>
<tr>
<td>Long</td>
<td>10-11</td>
<td>kepkvvfak</td>
</tr>
<tr>
<td>Long</td>
<td>11-12</td>
<td>tepkmnfak</td>
</tr>
<tr>
<td>Long</td>
<td>12-13</td>
<td>tepkgvfab</td>
</tr>
<tr>
<td>Long</td>
<td>13-14</td>
<td>tepkavfak</td>
</tr>
<tr>
<td>Long</td>
<td>14-15</td>
<td>sepkavfak</td>
</tr>
<tr>
<td>Long</td>
<td>15-16</td>
<td>aepkvvfak</td>
</tr>
<tr>
<td>Long</td>
<td>16-17</td>
<td>aepkavfak</td>
</tr>
<tr>
<td>Long</td>
<td>17-18</td>
<td>aepkvvfak</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>18-19</td>
<td>epqise</td>
</tr>
<tr>
<td>Short non-proline containing</td>
<td>19-20</td>
<td>aak</td>
</tr>
<tr>
<td>Short non-proline containing</td>
<td>20-21</td>
<td>srvvk</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>21-22</td>
<td>eapvl</td>
</tr>
<tr>
<td>Short non-proline containing</td>
<td>22-23</td>
<td>emrqvr</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>23-24</td>
<td>telpvs</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>24-25</td>
<td>kelpvt</td>
</tr>
<tr>
<td>Short non-proline containing</td>
<td>25-26</td>
<td>eeqsit</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>26-27</td>
<td>sdipvv</td>
</tr>
<tr>
<td>Short non-proline containing</td>
<td>27-28</td>
<td>earevt</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>28-29</td>
<td>rvkpvv</td>
</tr>
<tr>
<td>Short non-proline containing</td>
<td>29-30</td>
<td>hdlhvg</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>30-31</td>
<td>rerpaa</td>
</tr>
<tr>
<td>Short non-proline containing</td>
<td>31-32</td>
<td>anc</td>
</tr>
<tr>
<td>Short non-proline containing</td>
<td>32-33</td>
<td>qgrrvh</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>33-34</td>
<td>tekpsv</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>34-35</td>
<td>harpvr</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>35-36</td>
<td>talpaq</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>36-37</td>
<td>rpmpah</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>37-38</td>
<td>kalpak</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>38-39</td>
<td>ralp</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>39-40</td>
<td>ralp</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>40-41</td>
<td>rampsk</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>41-42</td>
<td>ralp</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>42-43</td>
<td>rapqqpv</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>43-44</td>
<td>taapvr</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>44-45</td>
<td>rvprpk</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>45-46</td>
<td>tepevt</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>46-47</td>
<td>rapevt</td>
</tr>
<tr>
<td>Short non-proline containing</td>
<td>47-48</td>
<td>taknt</td>
</tr>
<tr>
<td>Short non-proline containing</td>
<td>48-49</td>
<td>rgwrle</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>49-50</td>
<td>lglpd</td>
</tr>
<tr>
<td>Short proline containing</td>
<td>50-51</td>
<td>ppkpv</td>
</tr>
</tbody>
</table>

Multiple solution structures of individual obscurin Ig-like domains are already in the Protein Data Base (PDB; Table 2-2). Included in this set of structures are many that connect to neighboring domains via short proline-containing linkers (i.e. Ig34, Ig35, and Ig36 in full-length obscurin). However, only two published structures- Ig58 and Ig59- are connected with a non-proline linker. Therefore, in order to generate a more robust
data set for studying domain/domain motion, we first solved the solution structure of Ig57, a domain that connects to Ig58 via a non-proline linker. The heteronuclear single quantum coherence (HSQC) spectrum of Ig57 is well-dispersed, and every backbone peak was subsequently sequence-specifically assigned (Figure 2-1A). The resulting solution structure is of high quality, with more than 10 distance restraints per residue and no violations greater than 0.40 Å (Table 2-3, Figure 2-2). The 20 best structures overlay well with each other, with a backbone root-mean-square deviation (RMSD) of residues in the Ig-like fold being $0.681 \pm 0.061$ Å. The best structure, judged by having the lowest RMSD, shows Ig57 arranged into a typical Ig-like fold, with its two beta sheets arranged into a beta sandwich-like fold (Figure 2-1B).

Next, we constructed a series of dual Ig domains. Ig34/35 and Ig35/36 have short proline-containing linkers, and Ig57/58 and Ig58/59 have short proline-absent linkers (Figure 2-

Table 2-2:
PDB accession numbers of solution structure human obscurin Ig-like domains (from CAC44768)

<table>
<thead>
<tr>
<th>PDB code</th>
<th>Obscurin domain</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>2ENY</td>
<td>Ig27</td>
<td>2735-2825</td>
</tr>
<tr>
<td>2EDF</td>
<td>Ig28</td>
<td>2826-2915</td>
</tr>
<tr>
<td>2EDR</td>
<td>Ig34</td>
<td>3361-3449</td>
</tr>
<tr>
<td>2EDT</td>
<td>Ig35</td>
<td>3449-3537</td>
</tr>
<tr>
<td>2EDW</td>
<td>Ig36</td>
<td>3537-3630</td>
</tr>
<tr>
<td>2EDH</td>
<td>Ig37</td>
<td>3614-3713</td>
</tr>
<tr>
<td>2EDQ</td>
<td>Ig38</td>
<td>3713-3806</td>
</tr>
<tr>
<td>2EDL</td>
<td>Ig39</td>
<td>3801-3897</td>
</tr>
<tr>
<td>2MWC</td>
<td>Ig58</td>
<td>4342-4430</td>
</tr>
<tr>
<td>2N56</td>
<td>Ig59</td>
<td>4430-4519</td>
</tr>
</tbody>
</table>
All of the domains, individually, are fully assigned using multidimensional heteronuclear Nuclear Magnetic Resonance Spectroscopy (NMR). For each dual domain system, the resulting HSQC is almost exactly the sum of the individual domain HSQCs overlaid on top of each other (Figure 2-3B). This indicates the individual domains do not significantly interact with their neighbor, except at the extreme poles where the linker connects the two domains (Figure 2-3C). In addition, there was no evidence of peak splitting in any of the HSQC spectra, indicating these tandem domains are either in a single conformation, or else are in fast exchange between several different conformations. In all cases, the linker residues between two domains were exchange-
broadened out and could not be assigned, regardless of temperature (37 °C, 25 °C, and 10 °C). Additionally, no nuclear Overhauser effect (NOE) correlations were observed.

**Table 2-3: NMR-derived restraints and statistics of 20 NMR structures of wild-type Ig57**

<table>
<thead>
<tr>
<th>Rmsd from distance constraints (Å)</th>
<th>&lt;20&gt;</th>
<th>Best Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total</strong> (884)</td>
<td>0.021 ± 0.001</td>
<td>0.018</td>
</tr>
<tr>
<td><strong>Intraresidue</strong> (250)</td>
<td>0.005 ± 0.003</td>
<td>0.005</td>
</tr>
<tr>
<td>**Sequential (</td>
<td>i - j</td>
<td>= 1)** (250)</td>
</tr>
<tr>
<td>**Medium range (1 &lt;</td>
<td>i - j</td>
<td>&lt; 4)** (71)</td>
</tr>
<tr>
<td><strong>Long range</strong> (</td>
<td>i - j</td>
<td>&gt; 4) (241)</td>
</tr>
<tr>
<td><strong>Hydrogen bonds</strong> (72)</td>
<td>0.042 ± 0.007</td>
<td>0.039</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rmsd from exptl dihedral constraints (°)</th>
<th>&lt;20&gt;</th>
<th>Best Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>φ, ψ (112)</td>
<td>0.525 ± 0.140</td>
<td>0.538</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rmsd from exptl 13C chemical shifts</th>
<th>&lt;20&gt;</th>
<th>Best Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>13Cα (ppm)</td>
<td>1.350 ± 0.04</td>
<td>1.320</td>
</tr>
<tr>
<td>13Cβ (ppm)</td>
<td>1.740 ± 0.04</td>
<td>1.700</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rmsd from idealized geometry</th>
<th>&lt;20&gt;</th>
<th>Best Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bonds (Å)</td>
<td>0.004 ± 0.000</td>
<td>0.004</td>
</tr>
<tr>
<td>Angles (°)</td>
<td>0.565 ± 0.017</td>
<td>0.545</td>
</tr>
<tr>
<td>Impropers (°)</td>
<td>0.382 ± 0.021</td>
<td>0.342</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lennard-Jones potential energy (kcal/mol)</th>
<th>&lt;20&gt;</th>
<th>Best Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>-315 ± 8</td>
<td>77.6 ± 3.3</td>
<td>72.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>% most favorable region in the Ramachandran plot</th>
<th>&lt;20&gt;</th>
<th>Best Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>77.6 ± 3.3</td>
<td>72.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rmsd of the mean structure (Å)</th>
<th>&lt;20&gt;</th>
<th>Best Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>All backbone atoms</td>
<td>0.68 ± 0.06</td>
<td>0.55</td>
</tr>
<tr>
<td>All heavy atoms</td>
<td>1.26 ± 0.07</td>
<td>1.11</td>
</tr>
</tbody>
</table>

1 The 20 ensemble structures, <20>, are the results of simulated annealing calculations. The best structure is the closest to the average structure. The values shown for the <2> are the mean ± standard deviation.

2 None of the 20 structures has a distance violation > 0.35 Å or a dihedral angle violation of > 5°. The force constants used in the SA calculations are as follows: 1000 kcal mol⁻¹ Å² for bond length, 500 kcal mol⁻¹ rad⁻² for angles and improper torsions, 4 kcal mol⁻¹ Å⁻⁴ for the quartic van der Waals (vdw) repulsion term (hard-sphere effective vdw set to 0.8 times their values in CHARMM parameters), 50 kcal mole⁻¹ Å⁻² for experimental distance constraints, 1 kcal mol⁻¹ Å⁻² for distance symmetry constraints, 0.5 kcal mol⁻¹ ppm⁻² for the 13C chemical shift constraints, and 1.0 for the conformational database potential. The force constants (in kcal Hz⁻²) used for dipolar coupling restraints is 0.50.

3 Lennard-Jones van der Waals energies were calculated using CHARMM parameters and were not used in any stage of the structure determination.

4 PROCHECK was utilized to generate the Ramachandran plot.

5 Backbone calculations include C⁹, N, and C’ atoms. Only residues 3–91 are included since no long-range NOE correlations were observed for residues 1–2 and 92–104 (the expression tag).
between tandem domains or between domains and their adjoining linkers, supporting the notion that these domains are dynamic relative to each other, and that all short linkers, regardless of composition, experience significant intermediate-timescale ($\mu$s-ms) motions.

Due to the paucity of inter-domain NOE correlations, we cannot use traditional NMR methods to determine the conformation of these dual domains in solution. Therefore, we attempted two orthogonal techniques to better understand the solution
structures of these tandem domain systems: small angle X-ray scattering (SAXS) and residual dipolar couplings (RDC). Guinier plots of SAXS data (Figure 2-4A, Figures 2-5, 2-6) show all dual domain systems are extended in solution and have similar $R_g$ values regardless of linker composition (Figure 2-4B). We next fit our SAXS data to an ensemble of tandem domain models, each with different domain/domain angles (Figure 2-4C). In all constructs, the models that best fit the experimental data were almost fully extended,
in agreement with our Guinier analysis. However, the residuals of our best fits were non-random in most cases. Therefore, we re-fit the data using a two-state model: one extended conformation and the other compact (Figure 2-4D and Figure 2-7). These two-state models showed a better fit with the data, suggesting that all constructs are usually, but not exclusively, extended. Further fitting of more complex models yielded
Figure 2-5: Guinier and RDC analysis. A) Zoom in of Guinier regions of Ig34/35 with labeled R_g at two concentrations: 1 mg/ml (black) and 3 mg/mL (red). B) RDC experimental data on Ig58/59. C) Using PALES, The fit of each individual domain and the dual domain of Ig58/59 to experimental RDC data.

Figure 2-6: Guinier plot and Guinier region zoom in for each dual domain system at two concentrations. A) Ig35/36 at 1 mg/mL (black) and 3 mg/mL (red). B) Ig57/58 at 1.2 mg/mL (black) and 1.4 mg/mL (red). C) Ig58/59 at 2.5 mg/mL (black) and 3.0 mg/mL (red).
progressively better fits. As additional evidence of this apparent flexibility, RDC data on two of the tandem constructs (Ig35/36 and Ig58/59) show while the individual domains fit the data well, the data cannot be forced to fit any single dual domain model (Figure 2-5A/B). In sum, for every dual construct we tested, we conclude tandem Ig domains are relatively extended but can also exist in multiple conformations.

The finding that every tandem dual domain system is both extended yet flexible seems paradoxical. To address the problem of how these systems can simultaneously have this kind of structure and flexibility, we require high-resolution information of the various domain/domain interfaces. However, no NOE measurements exist between any of these regions. Additionally, this apparent domain/domain flexibility precludes x-ray crystallography analysis due to the potential of significant crystal packing artifacts. Therefore, we turned to molecular dynamic simulations (MD) in an attempt to find possible domain/domain or domain/linker interactions. All subsequent tandem domain models were first equilibrated for >50 ns, and the angle between the domains in solution was then measured over an additional 50 nanoseconds in triplicate (Figure 2-8A). In all simulations, each of the dual domain systems maintained a relatively extended structure on average, but the inter-domain angle varied widely, with a maximum change of orientation ~50-70 degrees. A global examination of these simulations suggests these extended conformations are the result of steric hindrance between the domains; neighboring domains with short linkers clash into each other if the angle between them is
less than \( \sim 120 \) degrees. These elongated but dynamic dual domain systems persist at least into the microsecond regime (Figure 2-9), and are in excellent agreement with our experimental data.

Closer examination of these MD trials showed that in all simulations, multiple residues at the domain poles participate in long-lived, stabilizing interactions with moieties in the linkers (Figure 2-8B). Once these interactions form, they usually persist for the duration of the simulation and are largely independent of domain/domain bending. To study these interactions in more depth, we next performed steered molecular dynamics.
simulations (SMD) on these systems, where the domain termini were moved apart at a constant velocity of 1 Å/ns. By elongating the dual-domain systems, SMD gives a more controlled setting to study how these putative inter-domain and domain/linker interactions respond to bend and stretch. SMD also simulates a physiologically reasonable timescale of stretch, and thus gives us insight into how obscurin may respond to stretch in the cell.

When a slightly bent dual domain system is stretched, the domains first straighten, yet many of the inter-domain and domain/linker interactions remain intact (Figure 2-10A/B). This is accompanied by the addition of either no or very little work to the system (Figure 2-10C). Only after the domains completely straighten does the linker begin to extend and these interactions begin to break (Figure 2-10D). Thus, these noncovalent interactions, originating on linker regions or on loops within the Ig domains, are both long-lived and flexible. The existence of such flexible interactions explains how dual domain
systems can simultaneously be extended and dynamic. Despite each construct having a different composition, all four sets of dual domain systems displayed this same behavior (Figure 2-11). As the domains are stretched further, increasingly more work must be added to the system until individual domains unravel. This kind of work-stretch profile occurs in all model constructs, and is reminiscent of other well-studied multi-Ig-domain systems.\textsuperscript{16,50} These domain-rupturing events present an oft-used cellular mechanism through which obscurin can resist large stretch forces.\textsuperscript{17}
Figure 2-10: Representative steered molecular dynamics analysis. A) Domain angle vs time graph of Ig34/35. B) Distance between the functional groups in one Ig34/35 simulation. In these measurements, distances of ~5-6 Å denote the distance of a hydrogen bond in this trace. C) Work vs. time graph of Ig34/35. D) Hierarchical model of obscurin extending with increasing stretch. Domains first straighten, followed by linker straightening, followed by domain unraveling.
Figure 2-11: (Previous page) SMD simulations for each dual domain system. A) Domain angle vs time graph of Ig34/35 (top), work vs time graph of Ig 34/35 (middle), and distance between residues of likely interactions vs time of Ig 34/35 (bottom) for three different SMD simulations. B) SMD data on Ig35/36, following the same organization as (A). C) SMD data for Ig57/58. D) SMD data for Ig58/59.

Discussion

The N-terminal majority of obscurin is composed of unique Ig-like and FnIII-like modular domains. Of the approximately 60 linkers that connect these domains, around 70% are 3-4 residues in length. Here, we study four representative short linkers. Dual-domain systems with proline-containing linkers and dual-domain systems with proline-absent linkers are equivalently flexible in solution. Domain/domain orientation tends to be around 160 ± 20°: almost fully extended. MD studies suggest these multiple orientations are of near-equivalent energies, and thus experimental high-resolution techniques are inadequate for studying this type of multi-domain dynamic system. Through extensive MD simulations and analyses we find, in all constructs, short linkers facilitate specific domain/linker and domain/domain interactions. These interactions occur predominantly on loops and other disordered regions of the protein, and can tolerate both moderate compression and stretch. While the exact bonds that form are inherently unique at every interface, each construct we have studied exhibits multiple examples of these interactions. The overarching conclusion is while short linkers facilitate such interactions, the regions containing these bonds are sufficiently flexible to allow significant domain motion. However there is a limit to this flexibility; when the domains bend excessively, the surfaces begin to bump into each other thus resisting further bending. Thus the existence of short linkers may be a mechanism in multi-domain
proteins to avoid unwanted domain/domain clamshell formation. Conversely, when two extended domains are pulled apart, interdomain bonds break well before the domains themselves rupture.

Previous structural studies of a similar system in titin concluded that short linkers, similar to those present in obscurin, lead to an extended conformation of Ig domains, and this conformation is maintained through a series of domain/domain and domain/linker non-covalent interactions. However, computational studies on these same systems suggest that consecutive domains are flexible relative to each other. Thus, the idea presented in this study, that short linkers in obscurin facilitate domain/domain and domain/linker interactions and these interactions can tolerate domain motion, reconciles longstanding discrepancies between experimental and computational work on the molecular flexibility of titin.

From the data gathered here, we created a simple model of how obscurin behaves in solution (Figure 2-12). In this model, we assumed the obscurin molecule is unhindered between the beginning and the end of its tandem Ig region (i.e. it participates in no target
binding in the middle of the molecule), the Ig region consists of 60 domains, each domain is 4 nm in length, and a two-domain system bends a maximum of 45 degrees away from 180 degrees. With these inputs, one can create a random walk trajectory (for example, see Figure 2-12A). Figure 2-12B shows a distribution curve of the distance between the termini of this model and suggests they will be, on average, around 76 nm apart from each other in solution. Of note, the input values can be altered, resulting in minor changes in the average termini distance (Figure 2-13). In this model, it is worth noting that the distance between termini range from 0 nm to ~239 nm. Given these constraints, and given the work that others have done on similar proteins, a reasonable model of this system is a worm-like chain model. Thus, with knowledge of the persistence length and contour length, we can calculate the entropic energy required to completely extend obscurin (to 239 nm), and we find this force is small: only around 28 J/mol. Further
Figure 2-13: Modeled average length of obscurin. A) The average obscurin length as a function of domain angle (in degrees). The number of links = (number of domains - 1) for each model. B) The average obscurin length as a function of link number. The plots represent what angle the linkers can bend. These models were generated in the XYZ dimensions.
separation of the termini, up to around 270 nm (or around 5 Å per linker), requires the flexible noncovalent bonds to break in order to fully extend each linker region. From our SMD measurements, this extension is associated with 1-10 kJ/mol of work per linker. This extension range is likely where obscurin behaves as a physiologically relevant molecular spring. Extension past 270 nm begins unraveling individual Ig-like domains, and requires a significant amount of work, likely in a manner reminiscent of how titin resists overextension. Thus, if obscurin links two distal targets at each termini, it will behave as a slack rope as long as those targets are less than 240 nm from each other. As the targets separate further, obscurin begins behaving as a spring, progressively resisting more force as the objects are moved farther apart from each other. This model presents obvious control points to tune such a system; adding additional anchor points to obscurin through interactions with domains in the middle of the protein, will correspondingly reduce the chain length and create a stiffer spring. Our model is overly simplistic; obscurin contains several regions of longer linkers (Table 2-1), and some tandem domains may more strongly interact with each other. Additionally, parts of the obscurin C-terminus are non-modular and other parts contain signaling domains, which our model does not take into account. Further research in these other obscurin regions will lead to a more refined model, and should provide more detailed insights into how obscurin behaves in the context of the myocyte.
Conclusions

Here we show obscurin tandem Ig-domains adopt an elongated orientation in solution. Despite staying moderately extended, the domains have a range of flexibility. This physical characteristic is brought about through the soft interface between neighboring Ig domains, and the interactions this interface creates. These interactions are postulated to help prevent self-association with neighboring domains. As a consequence of this elongated-yet-dynamic structure, obscurin does not significantly resist stretching force until the inter-domain linkers, and eventually the Ig domains themselves, begin to unravel. This hierarchical stretching profile allows for a simple model of obscurin flexibility.

Materials and Methods

Protein Isolation

All chemicals were ACS grade or higher and were purchased from Fisher Scientific, unless otherwise specified. Recombinant $^{15}$N, $^{15}$N-$^{13}$C, and unlabeled protein were purified after overexpression in *Escherichia coli* (BL21(DE3)) using pET24a vector system (Novagen, San Diego). All constructs were induced at 37°C with 100 $\mu$M IPTG at an $\text{OD}_{600} = 0.6$ and grown for additional 4 hours at 37°C. Cells were sonicated and centrifuged in a small amount of buffer containing 50 mM phosphate pH 8.0, 300 mM NaCl, 10 mM imidazole and 1 mM PMSF. The resulting cleared supernatant was passed over Ni-NTA His-bind Resin (Novagen). The column was washed
extensively with 50 mM phosphate pH 8.0, 300 mM NaCl, 10 mM imidazole buffer, and eluted with the same buffer plus 500 mM imidazole. Fractions containing the protein were then concentrated in 5000Da MWCO concentrators (Corning SpinX, Tewksburg, MA) and applied to a Sephadex G75 (Sigma; St. Louis, MO) size exclusion chromatography column in 50 mM Tris pH 7.5, 20 mM NaCl, 0.35 mM NaN₃ (G75 buffer). Pure protein, as determined by SDS-PAGE, was once again concentrated in a 5000Da MWCO concentrator.

NMR

All data for NMR experiments were collected on a 600 MHz Bruker Avance II spectrometer equipped with a TXI room temperature 5 mm probe with z axis pulse field gradient coils. NMR samples were either collected at 10°C (for Ig 57) or 10-37°C (for all other samples) in 20 mM Tris pH 7.5, 20 mM NaCl, 0.35 mM NaN₃, and 0.3-1.0 mM protein with 10% D₂O. For Ig57, we collected a 2D HSQC and standard ¹⁵N-edited triple resonance experiments including HNCACB, CBCA(CO)NH, HNCO, HN(CA)CO, C(CO)NH, HCCCONH, ¹⁵N-edited TOCSY, ¹⁵N-edited NOESY and ¹³C-edited NOESY, in as previously described. For other constructs, we collected 3D HNCACB, CBCA(CO)NH, HNCO, HN(CA)CO data along with 2D HSQCs. Most experiments were collected with 128, 64, and 1024 points in the T₁, T₂, and T₃ dimensions, respectively. NMR data were processed with NMRPipe, extended in the indirect dimension via linear prediction, and the resulting spectra were analyzed via Sparky. In all samples, all visible HSQC backbone shifts were assigned. Chemical shifts for the
obscurin Ig57 domain have been deposited in BMRB under the accession number 30514. Ig34, Ig35, and Ig36 chemical shift assignments were kindly provided by Dr. Ayako Nomura (Riken Structural Biology Laboratory, Japan).

**Structure Calculation**

Interproton distance constraints were derived from 3D NOESY experiments (\(^{15}\text{N}\)-edited and \(^{13}\text{C}\)-edited 3D NOESY) as described previously.\(^{43}\) Dihedral constraints \(\psi \pm 20^\circ\) and \(\phi \pm 15^\circ\) for \(\alpha\)-helix and \(\psi \pm 40^\circ\) and \(\phi \pm 40^\circ\) for \(\beta\)-sheet were included based on TALOS+ and the chemical shift index of \(^1\text{H}\alpha\) and \(^{13}\text{C}\alpha\) atoms.\(^{55,56}\) Structural calculations were performed as described in References 23-24. Out of 200 structures, the final 20 were selected based on lowest Q-values and lowest RMSD from the average, and were of high quality based on the statistical criteria listed in Table 2-3. The overall backbone RMSD of ordered heavy atoms is 0.609Å. The coordinates of the human obscurin Ig57 structure have been deposited in the Protein Data Bank 6MG9.

**Residual Dipolar Coupling (RDC)**

Anisotropic IPAP experiments for RDC determination were performed using the same conditions as for the HSQC with the exception of using a stretched polyacrylamide gel.\(^{52,57}\) The gel was prepared using 4% acrylamide, and soaked with buffer prior to soaking with protein. RDC values were calculated using PALES software.\(^{58}\)
Small Angle X-ray Scattering (SAXS)

Different concentrations (1.0, 3.0, and 5.0 mg/mL) of various obscurin samples were prepared in the NMR buffer. SAXS data were collected at the 12-id-B beamlines of the Advanced Photon Source (Lemont, IL) as previously described. Guinier plots were created using Origin, and the radii of gyration of the protein constructs were calculated with the Guinier approximation. MultiFoXS was used to analyze the fit of SAXS and RDC data together, as well as to back-calculate the conformation that best fit the SAXS data.

Molecular Dynamics (MD)

All MD simulations were performed using the YASARA 12.4.1 software package, the Amber 03 force field, and explicit solvent (with 150 mM NaCl) in a box that extended 5 Å beyond the length of the extended construct at 37°C, and described in reference 43. All simulations were run for at least 50 ns in triplicate.

All steered molecular dynamics simulations were performed using the PMEMD module of the Amber 14 MD software package, using AMBERff12SB force field and in explicit solvent. For equilibrium simulations, a constant temperature of 300K was imposed using a Langevin thermostat with a collision frequency of 1 ps⁻¹. A constant velocity of 1.0 Å/ns (0.1 m/s) was used in order to simulate biologically relevant pulling forces. The SMD spring constant (rk2) was set to 0.2 and the temperature used was 310.0 K. Analysis was visualized using Gnuplot.
Obscurin Modeling

A mathematical model for obscurin was created using a 4 nm rod for each domain and nine degrees of freedom between each domain (135, 180, 225 degrees in the x, y, and z direction, along with diagonals). Rods are connected at random in one of the nine degrees of freedom. The total distance calculated is measure from the first rod to the final rod. The model was implemented using MATLAB. The WLC formula

\[ F \approx \frac{k_B T}{L_p} \left( \frac{1}{4 \left( 1 - r/L_c \right)^2} - \frac{1}{4} + \frac{r}{L_c} \right) \]

was used, where \( k_B \) is the Boltzmann constant, \( T \) is temperature in Kelvin, \( L_p \) is the persistence length calculated in MATLAB, \( r \) is the distance between the N and C termini of our model, and \( L_c \) is the fully extended chain (the contour length).
Appendix

SAXS Analysis Tutorial and Instructions

SAXS Analysis Instructions - through Guinier analysis.

**NOTE:** this requires the use of Origin, which is not a free program!! However, most schools have educational copies floating around. But just FYI: If you are doing this at home, you’ll need to first pay for this program.

Use Origin to analyze all files
This set of instructions is to analyze SAXS data. This guide will be not showing how to collect SAXS data, and begins with already collected data that has been mildly organized. Additionally, this guide is written for elongated systems, but the same analysis techniques apply. If the protein of interest is globular, only one Rg value needs to be calculated.

The files of interest should be called something like filename_all.dat and filename_av.dat

_all.dat files are all of the data for that sample
_av.dat files are the average of the data for that sample

**Naming convention Example:** for the data from the 1.1 mg/ml concentration of Ig3637, which was the 93rd sample ran, we named it 3637c1p1_00093. The _all.dat file should be 3637c1p1_00093_all.dat. The _av.dat file should be 3637c1p1_av.dat. The c stands for concentration and the p stands for “point”. Origin doesn’t work well with special characters like periods.

1. the _all.dat files need to be looked at individually to make sure that there are no outliers.
   **Importing Multiple Files:** Open Origin – Under the word “Image” in the tool bar, click the icon that looks like 123 with an arrow to two spreadsheets (shown below). This is called import multiple ASCII. (We will use this button to import every file in the future). Then navigate to where the data files are saved and open the _all.dat of 1 concentration of the sample you want to look at. To sort only by _all.dat files and have no _av.dat files show up, type *all* into the file name box. There should be around 3 concentrations per sample. Click on the file, click add file, and then Ok. A Dialogue Box should open, but nothing needs to be changed so click Ok.

   The file should load into Origin. Column A is the Q value and all other columns are the actual data. Highlight all columns except for column A by clicking on B and dragging to the end of the sheet. Once highlighted, there is a button in the bottom left of the window that looks like a diagonal line (shown below). Click it and a graph should open.
The graph should look like an L or a right angle (shown below), and the scale of the axes needs to be changed. Double click on the x axis and change the “type” from Linear to “Log10” and change “From” from $1E^{-11}$ to $1E^{-3}$ or so. Click Apply. In the same dialogue box on the left, click on “Vertical” to edit the Y axis. Change the “type” to “Log10” again and change “from” to about $1E^{-2}$. Click OK. In the area of about .01 to .1 on the X axis, look to see that all of the lines more or less take the same path (shown below). If there are any outliers that are drastically different from the rest, find which column it is and delete it. If nothing needs to be deleted, the _av.dat files are good to use. If something was deleted, a new _av.dat file needs to be made for that sample.

Do this for all concentrations of each sample. Then close the origin window.

2. All of the _av.dat files for the sample need to be loaded into Origin. Only add the av files of the same sample, but different concentrations.

Using the same method as above in a new window, add in every concentration to Origin. Find a file, click add file, find the next concentration, add file, etc. until all concentrations for that sample have been added. Next add the Buffer_av.dat file. To sort only by _av.dat files and have no _all.dat files show up, type *av* into the file name box. Use the
Buffer file that is the closest to the number on the sample files. It is ok to import more than one buffer.

Example – if you have 3637c1p1_00093_av.dat. and you have 3 Buffer runs named Buffer1_00085_av.dat, Buffer2_00090_av.dat, and Buffer3_00099_av.dat, load in Buffer2 because that is the buffer that was run closest to that sample. When all files have been added, click OK. The dialogue box should open, and under “Import Options” -> “Import Mode”, change it from “Replace Existing Data” to “Start New Columns.” Now all of the concentrations and buffers are loaded into one window. They are not labelled, but they are in the order that they were added, so make sure to write that down. Each file has 3 columns, so the first 3 columns are the first file you added, next 3 columns are the second file etc. The first column of each file is the same so you only need one of them. We will keep column A. The second column of each file is the actual data we want. The third column of each file is the standard deviation. These third columns are also unnecessary. To hide the unnecessary columns, highlight one starting from the first unneeded column and holding “control” click on the other unneeded columns. Right click on one of them, Go to Hide/Unhide Columns -> Hide. They should disappear from view. Now there should just be Column A which is the Q values, and one column for each of the files imported. If you want to you can name the columns by clicking in the “Long Name” yellow box. This makes it easier to keep track of which column is which sample and concentration. Graph the columns as before and change the scales as before. When done, the lines should be different in the region between .01 and .1. What you should notice is that the Buffer should be the lowest line, and the concentrations should increase from that (shown below). Example: Buffer<c1p1<c2p3<c3p5. If the graph shows the low concentration as higher than a higher concentration, they were either mislabeled or not actually the labeled concentration.

3. Once the concentrations are determined, the buffer needs to be subtracted from each concentration. From the graph, on the left side there is a small window showing the books/graphs/tables that are in the sheet (shown below). Double click on the book and the
table of values should appear again. More columns need to be added to do the subtractions. To do this, click “Column” on the toolbar and then “Add new columns”. Type how many you want to add (add enough for each sample concentration, not buffers. Ex. If you loaded 3 concentrations and 2 buffers, only add 3 new columns) and click OK.

Highlight one of the new columns, right click on it and click “Set Column Values”. A dialogue box will pop up and here you can type a formula for every cell in the column. Pick one of the concentrations and know which buffer needs to be subtracted from it. In the new box, type:

\[ \text{col}(x) - 0.995 \times \text{col}(y) \]

where \( \text{col}(x) \) is the concentration column and \( \text{col}(y) \) is the buffer column (the column letters are at the top of each). Click ok and the data should fill out the column. If most or all values in the new subtraction column are positive, everything is good and can move on. If a lot of the values are negative, get to the dialogue box again and change the .995 to something lower like .95 or .97. You can also graph that column (changing the axes like before) and seeing when the part of the line around .01 is relatively flat. The area to the left of this is not flat and very jagged, but that is ok. Once the best number is found, this graph can be deleted.

Do this for each of the remaining concentrations (the .995 value can be different for each one, as long as most of the values are positive). It might be helpful to label which column is which concentration minus buffer.

4. Make more new columns. These columns will be used to create the Guinier plot. The Guinier plot is the graph which gives Rg (radius of gyration) for the sample. Each concentration will have its own Guinier, and an extra column needs to be made for the new x axis, so add n+1 columns where n is number of concentrations. Example: if there are 3 concentrations, add 4 columns.

Label the first new column \( Q^2 \) and set its value to:

\[ \text{col}(A) \times \text{col}(A) \]

Then click the next column and set the value to:

\[ \ln(\text{col}(x)) \]
Where col(x) is the subtraction column of the concentration you want to graph. Do this for each of the remaining columns and concentrations. It might be helpful to label these as ln(concentration subtraction) so you know which one is which.

**Highlight the Q^2 column – right click, Set As, X.** This is very important and the data will not make sense if this is not done.

Then graph one of the ln columns, but instead of using line, use Scatter which is the button next to line (shown below). Do not change the axis types this time – it needs to be linear. Double click on the X axis and change “from” to -.002 and change “to” to .02. This is the Guinier plot for the chosen concentration.

5. Now that the Guinier plot has been made, the Rg values need to be calculated using the slope of 2 portions of the graph. **If the protein in the sample is globular, only find the slope of the first region.** In this experiment we knew the protein was extended, so we needed two Rg values.

The graph should look something like this

One of the portions that you need the slope from is the area between .010 and .015. These values can be changed, it should just be a relatively flat part on the right side of the graph. This will give you the length in 1 dimension. This is the only region necessary for a globular protein.
The other portion you will need to zoom in to see. To zoom: On the very far left hand side, there is an icon that looks like a magnifying glass with a + inside. Click on that and drag from about .000 to .003. This zoomed in part will give you the length in another dimension. To make the graph normal again, there is a magnifying glass under the other one with a – inside (shown below). Click that and it should go back to normal.

The easiest way to check the slope quickly is to: Click on “Gadgets” in the toolbar, Quick Fit, Linear. This gives a yellow box that can be moved to show the slope of the line in the box. For a dual domain, the 1st slope should be around 50-70. The second domain slope should be around 500-600. Don’t go below about .0001 to find slope here. Use the yellow box to find good straight places to take the slope at.
After finding where you want to take the slope, there is a more comprehensive way which will give the residuals of that area. To do this: On the left side where the magnifying glass is, there is a button that looks like 2 vertical arrows that are pointing at each other called “Data Selector” (shown below). Click this and drag the lines that appeared on the graph to the point where you want to collect the slope. Then click “Analysis” in the toolbar, Fitting, Linear Fit, Open Dialogue, OK. A table should appear showing the slope of the line you just made. That table can be deleted because it now is a part of the files on the left hand side.
To check the residuals on that line, go back to the book by double clicking on it. At the bottom of the sheet, there should be new tabs called FitLinear1 and FitlinearCurve1. Click on FitLinear1 and at the bottom there are some graphs called “Residual Plots” – double click on this. Don’t use the other tab. The upper left graph is the one to look at. If all of the red dots are randomly scattered above and below the line, that data is good (shown below). If the dots seem to follow a pattern such as a trough or a peak, the data is bad and needs to be deleted and done again. A different area of the Guinier plot should be found that has good residuals. Right click and delete the residual tab (you don’t have to delete this tab if you plan on using the residuals in a figure), right click and delete the table, and right click on the line on the Guinier and delete if necessary.

All things on the left hand side can be renamed to keep it organized by right clicking, and then Rename. To rename the residual tab in the book, right click, then click Name and comments and rename it that way.
An easy naming convention is concentrationD1 and concentrationD2 to keep the dimensions separate.

Next go back to the Guinier and zoom in to the other dimension and get the slope of that line, same process as before.

Repeat the Guinier plot making, slope finding and residual checking for all other concentrations. Record the slope for each dimension of each concentration.

6. After all concentrations slopes have been recorded, plug the slope into this formula:
   \[(\text{Sqrt}(\text{slope} \times 3)) = \text{Rg value}\]
   \[\text{Rg value} \times 2 = \text{Distance}\]

Record these values.

Repeat for any other samples.

Below is an example of a Guinier Plot. The pink areas represent where the slopes were taken. The slope values can be seen in the tables to the left of the graph.

Basic overview of how to get to Guinier plot.

Take the data in average form of each concentration.

Subtract the buffer from the concentration

Plot this graph q vs subtraction

Take the ln of the subtraction

Plot this graph q^2 vs ln(subtraction)
SMD Tutorial and Instructions

Steered Molecular Dynamic Simulation Instructions

make a folder: mkdir (name of folder)
enter into a folder: cd
list folders: ls
take contents of file and print to screen: cat
to move up a folder: ../
to remove a folder: rm
to make sure the simulation is running, type: top
to see how long it will take, type: cat mdinfo

copy a pdb from the computer to a supercomputer:
Open the desktop where the .pdb file is located on X11
to copy this .pdb file into the supercomputer type:
scp 3435.pdb (supercomputer IP address)

In order to set the conditions for the molecule we must use the program XLeap which will neutralize, set parameters, and put in either explicit/implicit solvent. This program will basically tell the atoms how to interact:

**AMBER:**

**EXPLICIT SOLVENT**

**make sure that you have removed the hydrogen atoms in the pdb file beforehand**
to delete hydrogens in YASARA: Edit > delete > Hydrogens
to delete hydrogens in pymol: type- "remove hydrogens"
to open amber (this will open AMBER 12SB forcefield):
type:
  xleap –x –f /$AMBERHOME/dat/leap/cmd/leaprc.ff12SB
type:
  model=loadpdb "Ig3536GW.pdb"

click:
  File> Load PDB File (click file and load pdb project that you want to load)

in order to neutralize, type:
  charge model (this will tell you the charge)
  addions model K+ (# ions: how ever many to neutralize)
  addions model Cl-

in order to put in explicit solvent, type:
  solvateOct model TIP3PBOX 30.0
in order to set parameters, type:

    loadamberparams frcmod.ionsjc_tip3p

to save files that you created, type:

    saveamberparm model modelname.prmtop modelname.inpcrd

FYI \textit{inpcrd- input coordinates}
input prmtop => Amberfile
    inpcrd => Amber7 restart file

close AMBER by typing quit

\textbf{IMPLICIT SOLVENT}
to open amber (this will open AMBER 12SB forcefield):
type:

    xleap -x -f $AMBERHOME/dat/leap/cmd/leaprc.ff12SB

type:

    model
click:

    > File> Load PDB File (click file and load pdb project that you want to load)

    to set the default PBRadii mbondi3
    saveamberparm model model name.prmtop modelname.inpcrd

close AMBER by typing quit

\textbf{Running the Simulation}
\textbf{IMPLICIT SOLVENT:}
to run the stretch simulation requires 3 steps:

\begin{enumerate}
\item Minimize structure
\item Equilibrate
\item Pull
\end{enumerate}

*these files will be created in emacs:

    some tips:
    make sure to hit enter after the backslash
ttb=0 implies implicit solvent
igb=8 gives the forcefield you are using (ff12SB)
cut=99 which tells the program how many angstroms away each atom can see
to save in emacs: control x, control s
to exit emacs: control x, control c
Minimization:

**type:**
```
emacs modelname.in (this is an input file)
```

**in emacs, type:**
```
modelname-equilibration
&cntrl
imin=1,
maxcyc=5000,
ncyc=500,
ntb=0,
igb=8,
cut=999,
/
```
```
exit emacs
```

the next step is to start the minimization, this requires a command:
```
pmemd.cuda -O -i modelname.in -o modelname.out -c modelname.inpcrd -p modelname.prmtop -r modelname-min.rst
```

If you want to see the molecule in VMD:

**type:**
```
vmd modelname.prmtop
```
```
in VMD: file- new molecule-
load prmtop file
file rst. file
file type: Amber 7 restart
```

Equilibration:
```
emacs modelname-equil.in
model name-equil
&cntrl
irest=0, ntx=1, ig=-1,
imin=0, nt=0,
igb=8, ntp=1000, nt=x=1000
ntt=3, gamma ln=1.0,
tempi=0.0, temp0=300.0,
nstlim=1000000, dt=0.002,
cut=100,
ntwr=2000,
ntc=2, ntf=2,
ioutfm=1,ntxo=2,
/
```
```
Save and quit
```
** If you are starting equil. for the first time (for that model), irest=0 and ntx=1. If restarting equil. irest=1 and ntx=5

emacs submit-script.sh (which has been copied into folder) at bottom:
the next step is to start the equilibration, this requires a command:

to see if equilibration is done:
    vmd modelname.prmtop
    file-new molecule
    prmtop.load .mdcrd file file type: NetCDF, load all at once
Extensions- Analysis- RMSD trajectory tool
protein (whole thing) or resid # to # (certain residues)
check backbone and plot
align RMSD

Pulling or compression:
emacs modelname-equil.in
    model name-equil
    &cntrl
    irest=1,ntx=5, ig=-1
    imin=0, ntb=0
    igb=8, ntp=1000, ntw=1000
    ntt=3, gamma_in=1.0
    temp=300.0, temp0=300.0
    nstlim=100000000, dt=0.001
    cut=999
    jar=1
    ntw=1000
    NTC=2, NtF=2
    ioutfm=1, nto=2

    &wt type='DUMPRED', istep=1000, /
    &wt type='END', /
    DISANG= dist.RST
    DUMPAVE = dist_vs_t
    LISTIN= POUT
    LISTOUT=POUT

write dist.RST file:
    For constant velocity:
    &rst iat=___, r2=___, r2a=___, rk2=____
iat is the two atoms which are being pulled- choose alpha carbon in VMD
r2 is the starting distance (vmd-hold 2 while clicking both atoms)
r2a is the final distance
rk2 is the spring constant

to run the simulation:

EXPLICIT SOLVENT:

To run stretch simulation:
5 steps-
1. Optimize structure (minimize)
2. Heating calculation
3. Density- NPT
4. Equilibration (2days)
5. Pulling (3-5 days)
   Scripts for each of these input files is written out below. Or copy it from an old file.
   some tips:
   make sure to hit enter after the backslash
   ntb=1 implies explicit solvent
   igb=8 gives the forcefield you are using (ff12SB)
   cut=99 which tells the program how many angstroms away each atom can see
   to save in emacs: control x, control s
   to exit emacs: control x, control c

3435minimization example

type emacs modelname-min.in

&cntrl
   imin   = 1,
   maxcyc = 1000,
   ncyc   = 500,
   ntb    = 1,
   cut    = 10.0
/

save this file: file>save as
pmemd.cuda –O –i modelname.in –o modelname.out –c modelname.inpcrd –p modelname.prmtop –r modelname-min.rst &

**to make sure the simulation is running, type: top
to see how long it will take, type: cat mdinfo

3435Heating example

type emacs modelname-heat.in

3435heating
&cntrl
imin = 0,
irest = 0, ig=-1,
ntx = 1,
ntb = 1,
cut = 10.0,
ntc = 2,
tf  = 2,
tempi = 0.0,
temp0 = 300.0,
ntt = 3,
gamma_ln = 1.0,
stlim = 10000, dt = 0.002
ntpr = 100, ntwx = 100, ntwr = -500
ioutfm=1, ntxo=2
/

pmemd.cuda -O -i 3536heat.in -o Ig3536heat.out -c Ig3536-min.rst -p Ig3536-small3.prmtop -r Ig3536heat.rst &

Density example

type emacs modelname-density.in

3435density
&cntrl
imin = 0, irest = 1, ntx = 5,
ntb = 2, pres0 = 1.0, ntp = 1,
taup = 2.0,
cut = 10.0, ntr = 0,
ntc = 2, ntf = 2,
tempi = 310.0, temp0 = 310.0,
ntt = 3, gamma_ln = 1.0,
nstlim = 50000, dt = 0.002,
ntpr = 500, ntwx = 500, ntwr = 1000,
ioutfm=1,ntxo=2
ig=-1
/

pmemd.cuda -O -i Ig3536density.in -o Ig3536density.out -c Ig3536heat.rst_500 -p Ig3536-small3.prmtop -r Ig3536density.rst -x Ig3536density.nc &

Equilibration example

type emacs modelname-equil.in

Ig3536-equil.in
&cntrl
  imin = 0, irest = 1, ntx = 5,
  ntb = 1,
  cut = 8.0, ntr = 0,
  ntc = 2, ntf = 2,
  tempi = 310.0, temp0 = 310.0,
  ntt = 3, gamma_ln = 1.0,
  nstlim = 10000000, dt = 0.002,
  ntrpr = 2000, ntwx = 2000, ntwr = 2000,
ioutfm=1,ntxo=2,
ig=-1
/

pmemd.cuda -O -i Ig3536-equil.in -o Ig3536-equil.out -c Ig3536density.rst -p Ig3536-small3.prmtop -r Ig3536-equil.rst -x Ig3536-equil.mdcrd &

Pulling or compression example

First make the dist.RST file:

FOR VMD- open vmd by typing “vmd modelname.prmtop”
load files for prmtop (i.e. vmd test.prmtop)
In VMD, open the mdcrd file
Use NetCDF
Change stride to 10 and load all at once
To run ptraj scripts, you also need this .RST file. Change the atoms you are pulling, the distance between the atoms, and the speed of pulling:

```
dist.RST
&rst iat=13,2690, (this number needs to be changed)
r2=84.36, (this number needs to be changed)
r2a=184.36, (r2+100)
rk2=0.2,/
```

Pulling or compression:

```
type emacs modelname-equilibration
   model name-equil
   &cntrl
   irest=1,ntx=5, ig=-1
   imin=0, nt=1,
   igb=0, ntr=1000, ntwx=1000
   ntt=3, gamma_ln=1.0
   temp=300.0, temp0=300.0
   nstlim=10000000, dt=0.002
   cut=8
   jar=1
   ntwr=1000
   NTC=2, NtF=2
   ioutfm=1, ntxo=2
   /
   &wt type='DUMPREQ', istep=1000, /
   &wt type='END', /
   DISANG= dist.RST
   DUMPAVE = dist_vs_t
   LISTIN= POUT
   LISTOUT=POUT
```

To run pulling:

```
```

to analyze your data, use these ptraj scripts:

```
ptraj.script.distance
trajin 3435-small-pull.nc
```
distance l1dist :86 :91 out l1dist.dat
distance d1dist :5 :85 out d1dist.dat
distance d2dist :92 :174 out d2dist.dat

cpptraj modelname.prmtop <ptraj.script.distance> distance.out

**ptraj.script.angles**
trajin 3435-small-pull.nc
angle d1d2ang :5@C,CA,N :86-92@CA,C,N :174@CA,C,N out d1d2ang.dat mass
dihedral d1d2dihed :5@C,CA,N :77@CA,C,N :106@CA,C,N :163@CA,C,N out d1d2dihed.dat mass

cpptraj modelname.prmtop <ptraj.script.angles> angles.out

**ptraj.script.interactions**
trajin 3435-small-pull.nc
#Domain-linker contacts
nativecontacts :5-85 :86-91 writecontacts native-d1l1.dat resout nativeres-d1l1.dat distance 8.0 out native-d1l1.out first name native-d1l1 byresidue
nativecontacts :86-91 :92-174 writecontacts native-d2l1.dat resout nativeres-d2l1.dat distance 8.0 out native-d2l1.out first name native-d2l1 byresidue

#Domain-Domain Contacts
#quit
nativecontacts :5-85 :92-174 writecontacts native-d1d2.dat resout nativeres-d1d2.dat distance 8.0 out native-d1d2.out first name native-d1d2 byresidue
#map mapout nativemap.dat series seriesout nativeseries.dat
#run
#runanalysis lifetime native1[nonnative] out lifenative.dat nosort

cpptraj modelname.prmtop <ptraj.script.interactions> interactions.out

EMACS
Save: control x, control s
Exit: control x, control c

to get plots
gnuplot
plot 'dist_vs_t' u 0:4 w l
plot 'd1d2dihed.dat.mass' u 1:2 w l
etc.
find storage things

ls -ltrh */* | grep ‘[0-9]G’ - lists all files more than 1 gig
  ls = list
  l = longform
  t = timestamp
  r = reverse timestamp
  h = human
  grep = search
  */* = all directories

du -sh directory - how the storage is used.
  du = disk usage
  s = search
  h = human

To convert a .nc file to a .pdb file
  Type emacs nc_convert.in
  trajin filename.nc
  strip :WAT
  strip :K+
  strip :Cl-
  trajout filename.pdb

save this file

then type cpptraj filename.prmtop nc_convert.in
References


20. Borzok MA, Catino DH, Nicholson JD, Kontrogianni-Konstantopoulos A, Bloch RJ


43. Rossi D, Palmio J, Evilä A, Galli L, Barone V, Caldwell TA, Policke RA, Aldkheil E,


Disulfide Bonds within BST-2 Enhance Tensile Strength during Viral Tethering.


