Native Ion Mobility Mass Spectrometry: Characterizing Biological Assemblies and Modeling their Structures

Samuel Tabor Marionni

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2015

Reading Committee:

Matthew F. Bush, Chair

Robert E. Synovec

Dustin J. Maly

James E. Bruce

Program Authorized to Offer Degree:
Chemistry

University of Washington

Abstract

Native Ion Mobility Mass Spectrometry: Characterizing Biological Assemblies and Modeling their Structures

Samuel Tabor Marionni

Chair of the Supervisory Committee:
Assistant Professor Matthew F. Bush
Department of Chemistry

Native mass spectrometry (MS) is an increasingly important structural biology technique for characterizing protein complexes. Conventional structural techniques such as X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy can produce very high-resolution structures, however large quantities of protein are needed, heterogeneity complicates structural elucidation, and higher-order complexes of biomolecules are difficult to characterize with these techniques. Native MS is rapid and requires very small amounts of sample. Though the data is not as high-resolution, information about stoichiometry, subunit topology, and ligand-binding, is readily obtained, making native MS very complementary to these techniques. When coupled with ion mobility, geometric information in the form of a collision cross section (Ω) can be obtained as well. Integrative modeling approaches are emerging that integrate gas-phase techniques — such as native MS, ion mobility, chemical cross-linking, and other forms of protein MS — with conventional solution-phase techniques and computational modeling. While conducting the research discussed in this dissertation, I used native MS to investigate two biological systems: a mammalian circadian clock protein complex and a series of engineered fusion proteins.

Cryptochromes share a great deal of homology with DNA photolyases and are known to act as blue-light photoreceptors in plants and insects, however their role in regulating

the circadian clock in mammals is less understood. Native MS was used to show that flavin adenine dinucleotide (FAD), a cofactor in plant and insect cryptochromes, has comparatively weak binding to the mammalian cryptochrome mCRY2. Further, it was found that the full-length of mCRY2 is prone to degradation in solution, however its photolyase homology domain (PHR) is quite stable in solution. Subsequent crystallization of the PHR showing an open FAD binding pocket supported these native MS observations. Native MS of mCRY2 complexed to the ubiquitin ligase proteins FBXL3 and SKP1 (CRY2–FBXL3–SKP1) revealed that complex has two conformational populations in solution. Ion mobility data identified one of these conformers corresponds to the crystal structure. Two modeling approaches were used to characterize the second, more extended conformer. First, residues missing from the PDB structure were reconstructed *in silico* and calculated Ω values were compared with the ion mobility data. Secondly, the complex was analytically decomposed into mobile domains. The structure was pivoted at the interface of these domains to generate an ensemble of 54,000 structures. For each structure, Ω values and steric clashes were calculated. These data were integrated with previously reported cross-linking data using a scoring function, and it was found that both the CRY2 and FBXL3 subunits are likely flexible and are key components in the conformational switch.

The type II secretion system (T2SS) is a large molecular machine that Gram-negative bacteria use to secrete fully-folded protein products into extracellular space. The structure of the T2SS is of considerable biological interest, however few subcomplexes have been characterized structurally as neighboring subunit interactions are necessary for oligomerization. Fusion protein complexes were created to assist in oligomerization, however the dynamics of the complexes were inconsistent and heavily dependent on the fusion strategy. Native MS was used to rapidly characterize these complexes, revealing mixed stoichiometries and co-purifying proteins forming complexes with the fusion proteins. Using in-house software (NativeFit) and novel charge reduction technology, cation to anion proton transfer reactions

(CAPTR), it was possible to quantify relative amounts of mixed stoichiometries and identify co-purifying proteins.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

First and foremost I want to thank my mother, Joan, for raising me and providing an education and supportive ear. Sometimes when grad school gets to be too much, you just need to hear that in the grand scheme of things, small failures don't matter and our happiness comes first. Also my sister, Rorie, for teaching me that we need to ignore the perfectionism to which we are so inclined and remember our purpose, whatever that may be. My big sister taught me that a fulfilling career should involve helping others, and that passion and purpose will follow.

The last stretch of grad school was also much brighter after my partner, Solomon, came into my life. He is the love of my life, my closest confidant, and has heard the lion's share of my gripes not only about grad school but also pretty much everything else that pops into my mind (I tend to gripe a lot). Anyone who can stay involved with a grad student while he's writing his dissertation deserves a medal.

Many thanks to all of the members of my cohort, many of whom have beaten me out of here and are going on to do some great things. I'd especially like to thank Jen Brookes, my coffee buddy and a truly great friend the past six years. She is someone I respect a great deal, who not only overcame adversity but came out on top. I am incredibly proud to be her friend and continue to be inspired by her professionally as well.

I would like to thank all the members of the Lalic lab for creating a great environment while I had the pleasure of working with them. I've learned a lot from these people and had a great time working with all of them. I'd especially like to acknowledge Aaron Whittaker for training me and sharing a workspace as well as Mycah Uehling, a member of my cohort and a collaborator on a project (a successful one, I might add). I'd also like to acknowledge

Hester Dang, Richard Rucker, and Nick Cox, all of whom were great colleagues to have during my time in the lab. Many thanks especially go to my previous advisor, Gojko Lalic, who got me started in grad school, taught me a lot not only about chemistry but also strong work ethic and scientific integrity, and supported me greatly during my transition to a new research group.

I want to thank all the members of the Bush lab with whom I've worked over the past 4+ years. I have to give a shout to Sam Allen, the only other grad student for the first year of the group, helping to earn our lab the nickname "the Sams" and making sure we got launched successfully. This guy works harder than anyone I know and it's for sure been paying off the past few years. To Ken Laszlo, the one who ruined our naming scheme but was a welcome addition to the lab nonetheless. I'd like to thank Kim Davidson for sharing an office with me and carrying the IPython Notebook torch after I leave. To Rae Eaton for frequently providing baked goods and being courageous enough to take over after all of us old folks leave the lab. And of course all the previous members of the lab I've had the pleasure of knowing: Tracy Stanzel, Myung Cha, Alicia Schwartz, Chrissy Stachl, Nora Munger, Cindy Wei, and Stephanie Heard.

I'd of course like to thank my advisor, Matt Bush. Matt took a chance when he accepted a second-year grad student switching from another group, but never made me feel as if I were somehow at a disadvantage because of this. Matt gave me some really great projects to work on during my time here and gave me the flexibility to explore my interests within those projects, ultimately allowing me to guide myself down the path I now find myself on. Matt has guided and supported me, helped me find a postdoc position and take the next step, and ultimately helped me get to this very point where I can now say that I accomplished what I set out to do.

I'd like to thank all the other faculty and staff in the Department of Chemistry with whom I've worked. I'd like to acknowledge Martin Sadilek, who runs a fantastic mass spec

facility here in the chemistry department. Martin has helped greatly not only with teaching the students about mass spec, but has also helped me with my own experiments and taught me a great deal about mass spec. I'd also like to thank Tom Leach, with whom I've had the pleasure of working for several years now. I'd also like to thank Kim Quigley and Diana Knight who are always there to help put out the fires which inevitably show up. I'd also like to thank Krista Holden, our graduate program coordinator, for helping to make my transition out of the department seamless, as well as our previous graduate program coordinators Ashley Zigler and Lisa Nordlund. I'd like to thank all of my committee members, Dustin Maly, Rob Synovec, and Jim Bruce, as well as previous committee members Forrest Michael, Mike Heinekey, and Dan Eisenberg.

I'd also like to thank my collaborators on various projects, Ning Zheng, Wim Hol, Weiman Xing, Connie Lu, Stewart Turley, and Young-Jun Park. I'd like to thank Priska von Haller at the UW Proteomics Resource (UWPR) for help with experiments and guidance for getting our LC-MS system online, as well running a great facility and giving me the opportunity to present at the UWPR symposium.

Finally, I'd like to thank all of the great friends outside of the university who I've met over the past six years. There are too many to name, but suffice it to say I could have never made it through the past six years without a circle of friends as caring and loyal as the one that has surrounded me.

DEDICATION

in memory of my father, Paul

Chapter 1
INTRODUCTION

1.1 Native Mass Spectrometry

Mass spectrometry (MS) is a well-established technique that has proven particularly suitable for investigating biological systems. Once limited to small molecules, the advent of soft ionization methods, particularly electrospray ionization (ESI), has enabled the analysis of larger biomolecules such as proteins and protein complexes [1–3]. Conventional protein mass spectrometry focused on the analysis of peptides and denatured intact proteins to obtain information about sequence and post-translational modifications (PTMs) [4, 5]. In traditional protein MS, proteins are denatured and solubilized in an acidic organic/aqueous solution prior to electrospray, disrupting noncovalent interactions and forming highly-charged ions during electrospray due to the increased analyte surface area [6]. These highly charged monomers are detected at low m/z ranges where most mass spectrometers are most accurate and sensitive, making them suitable both for determining highly accurate monomer masses, as well as for peptide identification and sequencing, especially when performed as tandem mass spectrometry experiments.

Interest has grown in using nanoelectrospray ionization (nESI) coupled with nondenaturing solution conditions, such as aqueous ammonium acetate buffers at biologically relevant ionic strength and pH, to transport fully folded protein complexes into the gas phase with minimal disruption to their native conformation [7] and noncovalent interactions [8–15]. Analytes can range in size and complexity, from single-subunit proteins such as ubiquitin (8.5 kDa) [16] to 18 MDa virus capsids [17]. As a structural technique, native mass spectrometry is advantageous in that very little sample is needed and the structural information

obtained is complementary to that of X-ray crystallography, nuclear magnetic resonance (NMR), and cryo-electron microscopy (cryo-EM) [13, 18]. Native MS provides complementary structural information such as subunit stoichiometry and protein-ligand interactions from the analysis of these intact, native-like protein complexes [10, 13, 19]. Combined with subunit masses determined using conventional MS experiments, stoichiometry of subunits and ligand binding can be determined from the observed mass of intact protein complexes. Additional techniques such as collision-induced dissociation (CID) can be used to obtain information about subunit heterogeneity and arrangement [20,21], and to confirm charge-state assignments [22–24]. CID is described in further detail in Section 1.1.3. Native MS is well suited for complexes that are heterogeneous, cannot be crystallized, or exist in a low copy number [19]. Compared with other biophysical techniques, mass spectrometry is rapid and sensitive.

1.1.1 Electrospray Ionization (ESI) and Nanoelectrospray Ionization (nESI)

ESI is the ionization method used most often for the analysis of biological macromolecules such as protein complexes. In ESI, a voltage is delivered to a narrow capillary positioned at the front of the instrument. Flow of solution out of the tip of the capillary results in a cone-like shape, called a Taylor cone [25, 26]. At the very tip of this cone, a spray of small, charged droplets is formed. These droplets are attracted to the entrance of the instrument by the electric field. Desolvation of the droplet results in the analyte entering the instrument as a naked ion. The most accepted mechanism of desolvation for large, globular biomolecules is the charged-residue model (CRM) [27]. As solvent evaporates, the reduction in surface area of the droplet leads to an increase in Coulombic repulsion. As the droplets approach the Rayleigh limit [28, 29], they undergo fission events, ejecting smaller charged droplets of solvent [30,31]. Ultimately, through this combination of droplet fission and evaporation, the analyte is desolvated and enters the instrument as a naked ion.

Native MS has, in part, been enabled by the development of nano-ESI (nESI), which uses a tip size on the scale of 1–10 μm in diameter [11, 32]. nESI droplets are an order

of magnitude smaller than ESI droplets — about 100 nm in diameter [33]. nESI leads to greater ionization efficiency and reduces ion-suppression, leading to a reduction in m/z bias and more accurate determination of analyte ratios in mixtures [34]. Further, the likelihood of multiple analytes occupying the same droplet is also diminished at typical concentrations (~10 μM), which prevents formation of nonspecific oligomers [35]. nESI is most commonly carried out using gold-coated capillaries pulled to a small orifice [36], however it is also possible to achieve electrospray by inserting a non-reactive wire into the capillary such that it makes contact with the solution [37].

1.1.2 Instrumentation

Time-of-flight (TOF) mass analyzers are the most frequent choice of analyzer for native MS, with instrument modifications enabling mass measurement of non-covalent complexes at high m/z [38–41]. They are often coupled to a quadrupole mass filter on the front end (Q-TOF) to enable tandem MS experiments, but modifications are necessary to increase the m/z range of the quadrupole [39, 40]. Additionally, the time scale of TOF analyzers allows them to be coupled with ion mobility for the simultaneous analysis of all ions [42].

The Waters Synapt HDMS was the first commercially available mass spectrometer for native MS with ion mobility capabilities [43, 44], specifically a traveling-wave ion mobility (TWIMS) cell [45]. TWIMS cannot be used to measure collision cross sections directly, however analytes that have been previously characterized and measured using a drift tube may be used as calibrant ions [46, 47]. On our instrument, a second generation Synapt (Figure 1.1) [48], the TWIMS cell has been replaced with an RF-confining drift cell [49], enabling the direct measurement of collision cross sections of protein complexes.

1.1.3 Collision Induced Dissociation (CID)

After electrospray ionization, ions enter the vacuum environment of the spectrometer. On a Synapt G2 HDMS, the ions pass through a quadrupole mass filter and enter an argon-filled trap cell. The trap cell is a stacked-ring ion guide (SRIG) that radially confines ions

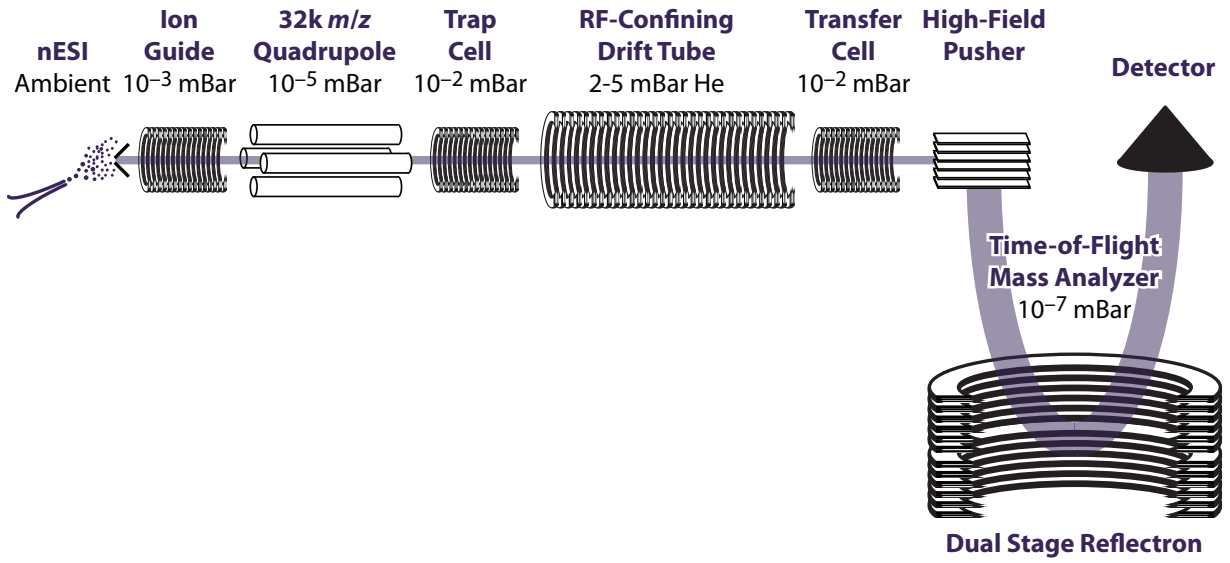


Figure 1.1: A Waters Synapt G2 HDMS in which the traveling-wave ion mobility cell was replaced with an RF-confining drift tube. Ions are generated with nESI and transmitted through the front of the instrument via a series of ion optics. The ions can be mass filtered in the quadrupole and injected into the argon-filled trap cell with varying amounts of energy to perform CID experiments. The next stage is the custom-built RF-confining drift cell that can be used for gas-phase separation and for the direct measurement of collision cross sections (Ω). After exiting the drift cell, the ions enter the transfer cell and are transmitted to the high resolution orthogonal TOF for mass analysis.

with RF. Collisions with the neutral bath gas molecules leads to a transfer of kinetic energy to internal energy, kinetically relaxing the ions and decreasing their momentum so that they can be confined by the RF and focused into a narrow beam. These collisions also further desolvate any residual adducts from solution [22, 50]. The injection voltage used can be adjusted to increase the level of desolvation. By increasing the injection energy and subsequent energy of the collisions further, kinetic energy is transfered into internal energy that results in ion heating [12, 51]. For protein complexes, this slow heating can cause smaller subunits to unfold, with the unfolding chain extending outward due to Coulombic repulsion [20, 52, 53]. This increased surface area leads to a migration of charges, increasing the Coulombic repulsion and unfolding. Eventually, a single subunit is ejected from the

complex, carrying a significant amount of charge with it [22]. The charge partitioning is asymmetric with respect to mass [24, 52], leading to the ejected monomer appearing at low m/z, whereas the charge-stripped oligomer appears at much higher m/z, decreased in mass but even more so in charge.

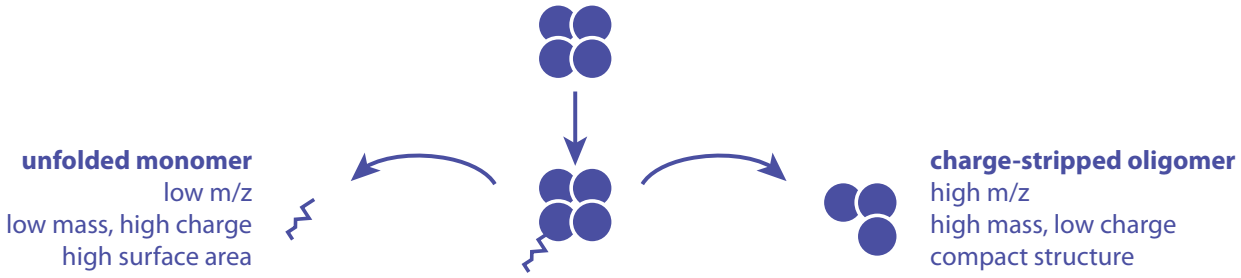


Figure 1.2: Cartoon of CID Mechanism.

Asymmetric charge partitioning associated with CID can be advantageous for structural determination. Appearance of the monomer CID product at low m/z regions, where the performance of the mass analyzer is much greater, combined with the isolation of a single subunit, allows the mass of the ejected monomer to be measured with high accuracy. In many cases, deviations from the sequence mass are observed, either due to unexpected truncations (Chapter 3) or posttranslational modifications (PTMs). This information can provide insight into global PTMs, assisting the researcher in the process of assigning charge states and fitting the spectra. Further, the accurate measurement of a single monomer mass also provides a very accurate mass fraction for that subunit in the larger complex, which can be indispensable for determining stoichiometries. In contrast, the higher m/z charge-stripped oligomer has reduced charge states that are spaced much farther apart, increasing the accuracy of the charge-state assignment. It is also possible to obtain subunit arrangement information, as external subunits often dissociate preferentially [20, 54].

1.2 Native Ion Mobility Mass Spectrometry (IM-MS)

Ion mobility (IM) is a gas-phase separation technique that separates analytes based on their size and charge [46, 55, 56]. Ions are injected into a drift cell containing a neutral carrier gas such as He or N_2, and an applied electric field accelerates the ions forward based on their charge. Collisions and interactions with neutral gas molecules retard larger species, increasing their drift time and resulting in separation on the millisecond timescale based on geometry. This timescale couples well with the rapid duty cycle of time-of-flight mass analyzers. In a traditional drift cell, the velocity of an ion (v) is the product of the electric field (E) and the mobility of the ion (K):

$$v = KE \tag{1.1}$$

Mobility can be used to determine the collision cross section (Ω), which can be approximated as the orientationally averaged cross-section of the analyte [57, 58]. Ω values are geometric parameters and provide more meaningful information about analyte structure [59]. More importantly, as geometric parameters they can provide structural information that is complementary to mass. Mobility values can be converted to Ω values using the Mason-Schamp equation [60]:

$$\Omega = \frac{3ez}{16N}\left(\frac{2\pi}{\mu k_b T}\right)^{1/2}\frac{1}{K} \tag{1.2}$$

where e is the elementary charge, z is the ion charge state, N is the drift-gas number density, μ is the reduced mass of the ion and the drift gas, k_B is the Boltzmann constant, and T is the drift-gas temperature.

1.3 Calculating Collision Cross Sections (Ω_{calc}) from Coarse Grain and Atomic Models

Proteins that are conformationally flexible in solution can become kinetically trapped in the gas phase [61], and consequently it is possible to probe solution dynamics with ion mobility.

Ω values of analytes are intrinsic properties and do not vary with instrumental setup or conditions (with the exception of the choice of bath gas and temperature). This low-resolution structural information complements the high-resolution data from X-ray crystallography and NMR. Using atomic structures, coarse-grain models, or a combination thereof, models of complex and heterogeneous protein assemblies can be constructed and their theoretical Ω values calculated. The calculated collision cross sections (Ω_{calc}) of these models can be compared with experimentally observed ion mobility data (Ω_{exp}), and consequently models can be constructed of complexes that would otherwise be challenging to characterize.

Multiple algorithms for calculating Ω values are available, but the most common are the projection approximation (PA) [62,63], exact hard-spheres scattering (EHSS) [64,65], and the trajectory method (TM) [57, 66]. The freely available and open-source program MOBCAL provides Ω_{calc} values for each of these three methods [57,64]. Of these methods, the trajectory method matches experimental values the best, but comes at a great computational cost. The merit of each of these approaches will be discussed below.

Briefly, the projection approximation creates a 2D projection, or "shadow", of the model at many angles [62], and is quite literally the orientationally-averaged projected cross section of the analyte and bath gas molecule ($r_{analyte}+r_{gas}$) from as many angles as possible until the value converges. It does not model atomic collisions and is therefore the most suitable method for coarse-grained models. The PA is the most computationally inexpensive of these methods, however it is notorious for underestimating the true Ω value as it does not consider the shape irregularities that are nearly ubiquitous in biomolecular complexes [58, 67]. A simple 2D approximation does not account for the effect of concavity or structural features such as pores and channels — specifically the effect of scattering angle or multiple scattering events — and does not take into account the long-range interactions between gas and ion. Owing to its simplicity and efficiency, many attempts have been made to circumvent this limitation, either by scaling the PA [9, 68], or expanding it to account for these irregularities [69–71].

The EHSS models the atomic collisions between the bath gas and the analyte [64, 65], treating all atoms as hard spheres and all collisions as elastic. By modeling atomic collisions,

the concavity and other shape features can be considered, as scattering angle and multiple scattering events influence the calculated values. Treating all collisions as being hard-sphere means that the model does not consider the polarizability of the gas or long-range potentials, however, and as such it is prone to overestimating Ω_{calc}.

Like the EHSS, the trajectory method considers collisions with the bath gas and the scattering angle, however it does not consider the collisions to be hard-sphere. Rather, the effects of long-range potential between the analyte and the bath gas are considered for determination of Ω_{calc} [57, 67]. The trajectory method is the superlative of the calculation methods, however the added calculation of potentials for every interaction adds considerable overhead, especially for large biomolecules and for large datasets involving many structures. For this reason, it will not be used for any of the analyses discussed in the next several chapters.

As previously mentioned, efforts have been made to scale lighter-weight calculations such as the PA to better approximate Ω_{exp}. Our group has taken advantage of a linear combination of the PA and EHSS, which has been shown to have very good agreement with experimentally obtained values [72]. The linear combination is as follows.

$$\Omega_{calc} = 0.84 \times \Omega_{PA} + 0.22 \times \Omega_{EHSS} \tag{1.3}$$

1.4 Structural Elucidation of Protein Complexes from Gas-Phase Data

Conventional solution-phase techniques such as X-ray crystallography [73, 74] and nuclear magnetic resonance spectroscopy (NMR) [75, 76] provide atomic resolution structures of proteins and protein complexes. These techniques are best suited for proteins that have a high copy number and are homogeneous, however many protein complexes of interest are transient in nature or have low abundance in the cell [77]. Additionally, many of these complexes are simply not amenable to crystallization, often crystallizing as lower-level oligomers. Because it can provide complete, albeit low resolution images of very large complexes, cryo-EM plays a significant role in integrative modeling approaches [78]. Chemical cross-linking can also

be integrated with native MS data to act as a spatial restraint when building models [79]. Clearly, gas-phase techniques such as native mass spectrometry and ion mobility have the potential to provide complementary information to bridge the gap between high-resolution structures of individual components and more complete models of high-level structures [80].

1.4.1 Stoichiometric Determination and Subunit Interactions

Perhaps the most advantageous use of native MS over conventional MS is the ability to probe the stoichiometry of noncovalent interactions. For homomers, the experimentally observed mass is expected to be some multiple of the monomer mass, which may be predicted from the sequence mass or determined using denatured MS or CID [81]. For heteromeric structures, stoichiometry can be more complex to assign, however combinations of masses can be winnowed to a few candidates, making mass accuracy of individual subunits of the utmost importance. Information about the arrangements of the subunits requires additional methods, as described below.

Subunit disruption can aid in the stoichiometric assignment of heteromers by simplifying structures to intermediate subcomplexes, as well as provide preliminary information about the interaction between neighboring subunits [82–84]. Disruption into smaller complexes in solution is performed by adding varying amounts of organic solvents such as methanol or DMSO [83, 85] or by increasing the ionic strength of the buffer [84, 85]. It is key to denature the solution just enough to cause partial disruption of the complexes without completely destroying all noncovalent interactions. The stoichiometry of these subcomplexes provides information about the disrupted interfaces and aids in the construction of larger interaction maps [84].

1.4.2 Integrative Modeling with Ion Mobility Data

If high resolution structures exist for a monomeric protein but little is known about higher-order oligomers, coarse-grained models can be built to represent various structures and subunit arrangements [7]. Coarse-grained models are generated by representing individual sub-

units as spheres. The radius of the sphere is based on the Ω_{calc} determined from an all-atom structure [46, 86] or Ω_{exp} for individual subunits, if available. Alternatively, an analogous technique can be employed generating bead models using size parameters from analytical ultracentrifugation (AUC) [78]. If a high-resolution structure does exist for a multisubunit structure but a coarse-grained model is desired, the collision cross sections of individual subunits are likewise calculated and then placed at the subunit's center of mass. An example of this approach is shown in Figure 1.3. These smaller components can then be assembled into larger complexes to model assembly and predict structure by varying the spacing and angles between these structures [87].

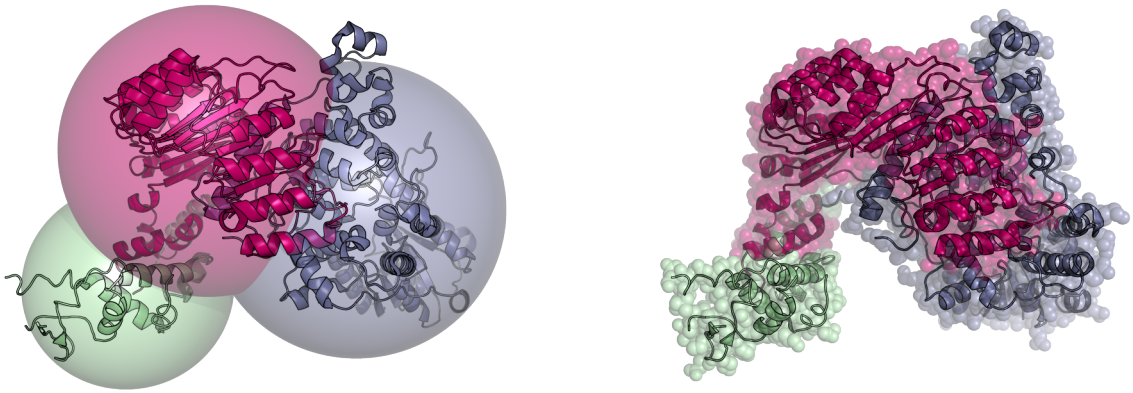


Figure 1.3: The coarse-grained (CG) model (left) of CRY2–FBXL3–SKP1 above represents each subunit with a single sphere, placed at that subunit's center of mass. The collision cross section of each subunit is calculated from the crystal structure using the linear combination described in Equation 1.3, and the radius of each sphere is adjusted to such that the cross section will equal that of the corresponding subunit Ω_{calc}. Using the IMPACT algorithm [68] to calculate the PA of both the CG and all-atom model, Ω_{calc} is found to be 13.7% larger in cross-section (Ω_{CG} = 66.3 nm^2, Ω_{atomic} = 58.3 nm^2). CG models best reflect all-atom models when subunits are globular, whereas FBXL3 (pink) is quite spiral in shape.

Models built by this approach need to be varied, with many possible arrangements considered, and evaluated using spatial restraints such as experimentally determined collision

cross sections [88, 89]. A coarse-grained approach was used by Pukala et al. to investigate potential topological models of ornithine carbamoyl transferase (PDB: 1PVV) and glutamine synthetase (PDB: 1HTO) [86]. Subunit disruption was used to analyze subcomplexes for stoichiometry, and from this information higher-order structures were constructed from coarse-grained spheres.

For large datasets of generated structures, it becomes necessary to evaluate candidates in a consistent and automatic fashion based on agreement with experimental data and other available parameters using a scoring function. Scoring functions are quite diverse and depend on the nature of the restraints in the system for which they are being implemented. Hall et al. used a Monte Carlo method to generate a population of 100,000 structures and developed a scoring function consisting of a summation of two harmonic penalty terms [88]. One term penalized sphere overlap by assuming a constant spherical packing density for globular proteins and penalizing decreases in volume. The second term similarly penalized structures whose Ω_{calc} deviated from the experimentally observed values. Politis et al. implemented a summation of three terms, describing connectivity restraints from native MS and cross-linking data with a third term to penalize overlap [90]. Given the diversity of such systems, it is important that such scoring functions be weighted to reflect the contribution of each constraint and for the functional forms chosen for these metrics to be chosen carefully [91]. Additionally, a software package exists to provide a platform and toolkit for designing such approaches [92].

1.5 Outline of Dissertation

For the remainder of this dissertation, the discussion will center around the role that native mass spectrometry has played in the characterization of two biological systems: a mammalian cryptochromes – ubiquitin ligase complex and fusion proteins of the type II secretion system.

I collaborated with two members of the Zheng Research Group, Prof. Ning Zheng and Dr. Weiman Xing, to characterize a mammalian cryptochrome both in isolation (mCRY2) and complexed to two proteins of its regulating ubiquitin ligase complex, FBXL3 and SKP1

(CRY2–FBXL3–SKP1). Native mass spectrometry results revealed that mCRY2 interaction with flavin adenine dinucleotide (FAD), a common cofactor for plant and insect cryptochromes, was very weak. This was supported by subsequent crystallization of mCRY2, revealing an open cofactor binding pocket. Native MS also revealed that full-length mCRY2(1–544) is prone to C-terminal degradation in solution, whereas a truncated form, mCRY2(1–512), is solution stable. Native MS of CRY2–FBXL3–SKP1 confirmed that the complex exists as a heterotrimer, with no evidence of posttranslational modifications observed.

Native MS also revealed that the charge-state distribution of CRY2–FBXL3–SKP1 is bimodal, suggesting two conformational populations. Ion mobility experiments were used to determine the collision cross sections of each charge state of CRY2–FBXL3–SKP1; it was determined that one of the conformational populations was geometrically similar to the previously described structure whereas the larger population had no model for comparison. Two approaches were used to model candidate structures for the extended conformer. The first approach involved reconstructing residues missing from the X-ray crystal structure (PDB: 4I6J) [93] with varying geometries, yielding a modest increase in cross section. Alternatively, the complex was analytically decomposed into mobile domains. The structure was pivoted at the interface of these domains to generate an ensemble of structures, and the collision cross sections of these structures were calculated. In addition, previously reported cross-linking data [94] for CRY2–FBXL3–SKP1 was incorporated along with a model for evaluating steric clashing in the simulated structures. A scoring function was used to evaluate these three parameters and identify promising structural motifs that could contribute to the conformational switch.

Native MS was also used to rapidly characterize fusion protein complexes of the type II secretion system. Engineered protein systems often fold unpredictably, and results from native mass spectrometry can be used to characterize these proteins and formulate strategies to improve fusion protein designs and expression systems. Spectra from these systems are often convoluted and difficult to interpret, and the observed masses may not resemble the masses predicted by sequence. Deconvolution of these spectra was achieved using in-house

developed software, and charge state assignments were confirmed using cation to anion proton transfer reactions (CAPTR), a charge-reduction technology developed within our research group. Using these tools, relative ratios of oligomers in mixed stoichiometries were measured and unintended binding partners were identified. Several fusion protein candidates that looked promising from native MS data were subsequently crystallized, validating the role of native mass spectrometry in characterizing these systems.

Finally, much of the Python code that was used in modeling and data analysis is included in the appendices of this dissertation in the form of Jupyter (formerly IPython) notebooks [95, 96]. Jupyter is a tool that allows Python code to be run interactively from within a web browser, creating a log of the analysis. The code can be annotated with formatted text, and the result is a cohesive narrative of the analysis that emerges — in essence, a laboratory notebook page for the data analysis. The notebook files are themselves executable, allowing others to easily reproduce the analysis or analyze their own data using the same work flow. They may also be exported as static text, as they have been for inclusion into this dissertation as appendices. I have included these notebooks as appendices as a means of removing the "black box" element of the computational methods. It is my hope that these appendices both validate my methodology and serve as a resource for those looking to implement the same modeling approaches in their own research.

Chapter 2

CHARACTERIZATION OF A MAMMALIAN CRY2 COMPLEXED TO SCF$^{\mathbf{FBXL3}}$

Portions of the work presented in this chapter have been published previously in

Weiman Xing, Luca Busino, Thomas R. Hinds, Samuel T. Marionni, Nabiha H. Saifee, Matthew F. Bush, Michele Pagano, and Ning Zheng. SCF$^{\mathrm{FBXL3}}$ ubiquitin ligase targets cryptochromes at their cofactor pocket. *Nature*, 496(7443):64–68, **2013**.

2.1 Abstract

Cryptochromes are circadian clock proteins that tightly bind FAD and act as blue-light photoreceptors in plants and insects, but they are believed to play a more complex role in mammals. The characterization of a mammalian cryptochrome, mCRY2, using native mass spectrometry (MS) is described herein. Native MS revealed that binding of FAD to the mCRY2 subunit is weak, which is consistent with the subsequent crystal structure showing an open binding pocket and fluorescence assays revealing a high K_d. Additionally, the full-length mCRY2 is vulnerable to proteolysis when in isolation, whereas the photolyase homology region (PHR) of mCRY2 is comparatively quite solution stable. Native MS also revealed that the complex forms a heterotrimer with two SCF proteins (CRY2–FBXL3–SKP1), with no apparent evidence of posttranslational modifications (PTMs). Our gas-phase results complement and compare well with other solution-phase techniques used to characterize mCRY2 and CRY2–FBXL3–SKP1. The mammalian cryptochrome mCRY2 has been successfully characterized both isolated and complexed to two proteins of the SCF$^{\mathrm{FBXL3}}$ ubiquitin ligase.

2.2 Background

Cryptochromes (CRYs) are circadian clock flavoproteins that are closely related to DNA photolyases and are found in a wide variety of plants and animals [97]. In animals, cryptochromes can be divided into type I and type II cryptochromes. Plant and type I insect cryptochromes act as blue-light photoreceptors [98], regulating growth and development in plants and modulating behavior and circadian rhythms in insects. Type II cryptochromes are generally found in mammals and play a significant role in circadian timing, but do so in a light-independent manner [98]. Structurally, mammalian CRYs have a conserved N-terminal domain that shares a great deal of homology with DNA photolyases, called the photolyase homology region (PHR), and a unique C-terminal extension that is not found in DNA photolyases [99].

The internal circadian rhythms are of great scientific and medical interest, as many of the body's metabolic processes display circadian fluctuations [100–102]. Metabolic syndrome is a growing global pandemic — as of 2008 it was estimated that between 20–30% of adults had some form of metabolic syndrome [103]. It is estimated that as many as 380 million people worldwide will have type 2 diabetes by 2025 [104]. While sedentary lifestyle and increased food consumption are key risk factors [105], considerable evidence links circadian rhythm disruption to these disorders [106]. Shift and nighttime workers, as a model for circadian rhythm disruptions, are at a higher risk of metabolic syndrome and other related conditions such as gastrointestinal disease, diabetes, cardiovascular disease, and increased cancer risk [107].

The degree of impact of the circadian rhythm on metabolism and nutrient processing is well established. Knockout studies on *Clock* and *Bmal1* genes in mice show impaired glucose tolerance depending on the affected tissue [101]; *Bmal1* knockouts in the pancreas cause hyperglycemia [108] whereas knockouts in the liver cause hypoglycemia [109]. Another study on the entrainment of wild-type mice to a 28 hour day showed increased blood glucose and insulin levels accompanied by overall decreases in insulin sensitivity [110].

To understand the critical role of CRYs, it is important to first consider the global hierarchy of the circadian rhythm in the body. In mammals, the master clock is a collection of approximately 20,000 cells located in the hypothalamus [111] called the suprachiasmatic nucleus (SCN). Each cell in the SCN is an autonomous clock, and these cells are coordinated both internally through the neurotransmitter GABA [112] and through external stimuli such as retinal signals in response to light [113]. In the absence of light/dark cycles, a ~24 hour rhythm will persist, however. Every cell in the body is regulated by it's own autonomous molecular circadian clock [114], however it is through the SCN that these peripheral clocks are synchronized throughout the body [115]. No synchronization amongst these peripheral tissues has been observed, and thus the phase alignment of all these clocks relies on the SCN [116, 117].

The intracellular clock mechanism is driven by a negative feedback loop [118]. In mouse models, the proteins CLOCK and BMAL1 form a heterodimer and drive the expression of 5 genes, the period genes *mPer1–mPer3* and the cryptochrome genes *mCry1* and *mCry2*. With the assistance of the mPER proteins [119], the cryptochromes travel back to the nucleus and inhibit the expression of *Clock* and *Bmal1*; this engenders the negative feedback loop. This cycle is crucial to the maintenance of the molecular clock. Various knock-outs of these genes lead to arrhythmic, lengthened, and/or shortened cycles [120–123]. One knock out study investigation the silencing of either the *cry1* or *cry2* genes revealed complementary behavior — knocking out *cry1* led to a shorter cycle whereas knocking out *cry2* led to a longer cycle [120]. This was only observed when mice were kept in total darkness; mice were able to sustain their circadian rhythm if exposed to regular light/dark cycles. When both genes were knocked out, the mice displayed total circadian arrhythmicity when raised in total darkness. This supports the role of cryptochromes in maintaining circadian rhythm in the absence of light/dark cues. Further, it has been shown that in knocking out the cryptochromes that transcription of both *mPer1* and *mPer2* is elevated [121]. However, in constant darkness no oscillation in the transcription of either gene was observed. This evidence points to the currently accepted hypothesis CRYs inhibit expression of *mPer* genes.

CRYs are degraded through the ubiquitin proteasome system [124]. One of the most well characterized ligase is the SKP1–CUL1–F-box complex (SCF), where the F-box family of proteins recognizes target proteins [125]. It has been shown that SCF^{FBXL3} recognizes CRYs for degradation [124]. Coimmunoprecipitation experiments with both wild-type and mutant FBXL3 showed that although all variants were able to interact with mutant versions of FBXL3, only wild-type FBXL3 could interact with CRY1, and binding to CRY2 was also significantly reduced. Further, nine other human F-box proteins failed to coimmunoprecipitate CRY1 and CRY2. Additionally, silencing FBXL3 was shown to inhibit the oscillation of *Clock-Bmal1* expression. Mice possessing a mutation (*After-hours*) in the *Fbxl3* gene were shown to have significantly longer circadian rhythm than the wild-type mice of the same population [126]. Another mutation (*Overtime*) in *Fbxl3*, when homozygous, resulted in a murine circadian rhythm ranging from 25–28 hours [127].

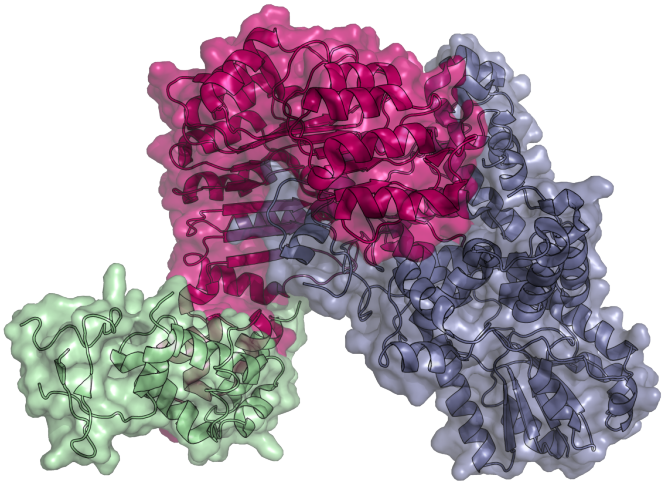


Figure 2.1: The crystal structure of CRY2–FBXL3–SKP1 (PDB: 4I6J), determined by Weiman Xing and Ning Zheng [93]. CRY2 is shown in deep blue, FBXL3 in pink, and SKP1 in pale green.

Herein, the native mass spectrometry results contributing to the characterization of

CRY2–FBXL3–SKP1 and elucidation of the crystal structure of CRY2–FBXL3–SKP1 (Figure 2.1) are discussed [93]. Specifically, native MS will be used to probe solution stability of the mCRY2 monomer and oligomerization of the heterotrimeric complex CRY2–FBXL3–SKP1. Additionally, the characterization of FAD-binding to the mCRY2 monomer will be explored using native MS.

2.3 Experimental Methods

Prior to analysis with native MS, mCRY2 solutions were exchanged into aqueous 1.00 M ammonium acetate using gel filtration chromatography. The fractions containing mCRY2 were then concentrated using a MicroSep centrifugal device (Pall Life Sciences, Ann Arbor, MI) with a 10 kDa molecular weight cutoff. The concentrated solutions were then diluted using the same buffer and re-concentrated three additional times. The CRY2–FBXL3–SKP1 sample was buffer exchanged into an aqueous 300 mM ammonium acetate solution at pH 8.0 using a Spin-X UF 500 μL Centrifugal Concentrator with a 10 kDa molecular weight cutoff (Corning Inc., Corning, NY). All experiments were performed using nanoelectrospray ionization and a Waters Synapt G2 HDMS mass spectrometer. Unless otherwise indicated, all spectra were calibrated externally using a solution of cesium iodide (16 mg mL^{-1} in 50/50 water/isopropanol).

2.4 Results

2.4.1 Solution Degradation of mCRY2(1–544)

A sample of mCRY2(1–544) (Figure 2.2a) was analyzed with native MS (Figure 2.2b, blue trace). Four peaks were observed with the maximum intensity at the peak corresponding to the +15 charge state. In most native MS experiments, charge state peaks are Gaussian in shape, however mCRY2 charge state peaks are wide and exhibit a structured, non-Gaussian shape. Upon zooming in on the peak corresponding to the +15 charge state (Figure 2.2c), multiple species are partially resolved contributing to the jagged peak shape. The mass

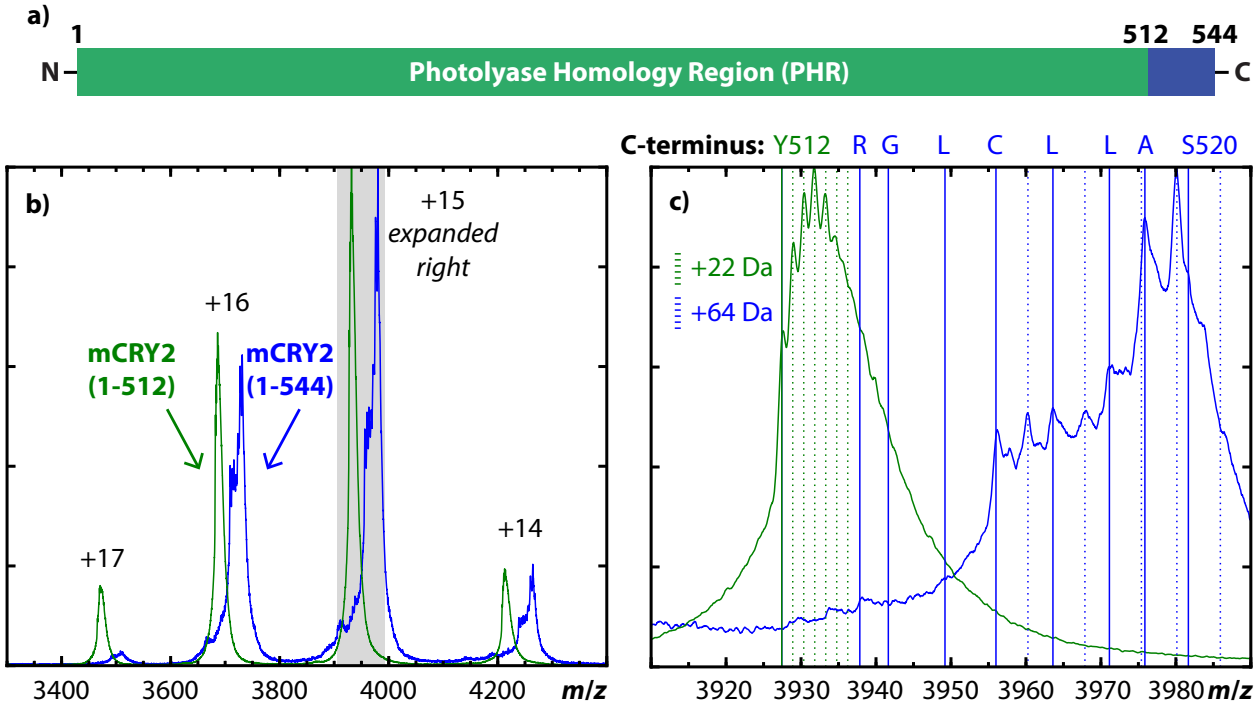


Figure 2.2: Native mass spectra of a mammalian CRY2 (mCRY2) prepared as amino acids 1–544 (blue) and 1–512 (green) are shown in (**b**). A sequence diagram is shown in (**a**). The region containing the +15 charged protein ions (grey box) is expanded in (**c**). Solid vertical lines show the expected m/z values for $(M_C+15H)^{+15}$ ions, where M_C corresponds the masses of various polypeptide chains terminated at different C-terminal positions, as indicated. The spectrum for mCRY2(1–544) indicates that this sample has undergone significant degradation, and that the C-terminal residues of the resulting proteins were predominantly C516, L517, L518, A519, or S520. Interestingly, a second series of peaks were also observed that had a mass ~64 Da greater than those of the assigned peaks and are marked by vertical dotted lines. The origin of that mass difference has not been determined. The spectrum for mCRY2(1–512) is consistent with the expected chain length. The additional structure in that peak is attributable to ions containing different numbers of sodium ions instead of protons (Na^+ - H^+ = 22 Da).

differences between these peaks reveal that each represents a different C-terminal cleavage product. Additionally, there is a series of peaks that are ~64 Da shifted in mass relative to the cleavage products (dotted blue line), however the source of this mass shift was not determined. Based on this result, Xing et al. expressed mCRY2(1–512) (PHR) and this

sample exhibited no evidence of degradation (Figure 2.2b, green trace). It was possible to resolve individual sodium adducts on the mCRY(1–512), accounting for much of the peak width, and no evidence of solution degradation was observed (Figure 2.2c). The observed mass of mCRY2(1–512) (m_{obs} = 58 899.0 Da) differed from that expected from the sequence (m_{seq} = 58 614.5 Da) by 284.5 Da, however this is not sufficiently strong evidence to suggest a cleavage or PTM as the spectrum was not calibrated with CsI.

2.4.2 mCRY2 Interaction with FAD

The native MS of the mCRY2 PHR interacting with FAD is shown in Figure 2.3. The sample (125 μM) was incubated in an ammonium acetate buffer containing 10 mM FAD for 1 hour at room temperature in order to facilitate saturation of the protein with FAD (ligand to protein ratio is ~80:1). This concentration of FAD is excessive for native MS and would be expected to result in non-specific adduction mimicking multiple binding, and thus the protein was buffer exchanged once into a buffer containing no FAD. The resulting mass spectrum showed that the *apo* form was dominant, whereas total saturation of the protein with persistent binding would be expected if mCRY2 had similar affinity to FAD as CRY2 from *Drosophila*.

Subsequent crystallization by Xing et al. of the mCRY2 PHR with bound FAD [93] supports the weak ligand binding observed by native MS (Figure 2.4). Further, Xing et al. subsequently determined the K_d to be approximately 40 µM by FAD fluorescence assay [93]. The crystal structure of the mCRY2 PHR [93] shows FAD bound in the predicted cofactor pocket (Figure 2.4, left). The structure reveals an open binding pocket, a marked contrast with those seen in plants and insects. mCRY2 was also crystallized without added FAD (PDB: 4I6E, not shown) and shows a similar structure, suggesting that the protein is stable without FAD and the binding pocket is indeed open. In contrast, in the crystal structure of *Drosophila* photolyase [128], FAD is seen deeply buried within the structure (Figure 2.4, right). For proteins acting as blue-light photoreceptors, a chromophore is critical to proper function [98], necessitating a tight binding pocket to retain FAD at all times.

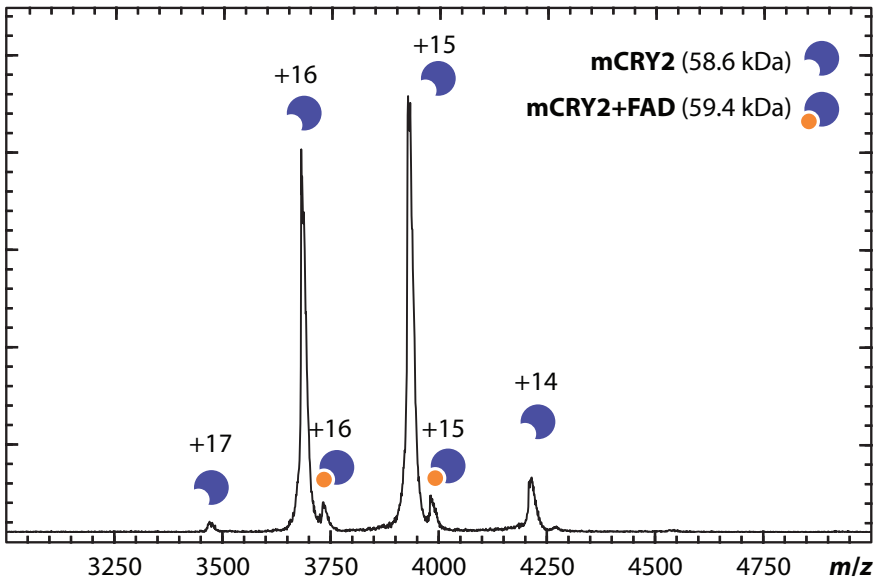


Figure 2.3: Native mass spectrum of mCRY2(1–512). A sample of mCRY2(1–512) was incubated for 1 hour at room temperature in the presence of 10 mM FAD. The sample was then buffer exchanged to remove unbound FAD and electrosprayed. The dominant species is unbound mCRY2(1–512) (m_{seq}=58.6 kDa, m_{obs}=59.0 kDa) with the FAD-bound species (m_{seq}=59.4 kDa, m_{obs}=59.7 kDa) appearing as a small peak with ~8% the intensity of the unbound form. The mass shift (Δm) between the two species is consistent with the mass of FAD (Δm=734.5 Da, m_{FAD}=785.56 Da).

Considering the weak binding of FAD in mammalian cryptochromes and the geometric and functional differences compared to plant and animal cryptochromes, the role of FAD and the cofactor pocket becomes less clear. This is consistent with previous studies indicating the ubiquitous presence of CRY2 in tissues shielded from light and the role CRYs play in the light-independent regulation of the clock [129].

Crystallization of CRY2–FBXL3–SKP1 has shown, in fact, that the open cofactor pocket of mCRY2 acts as a docking site for the C-terminal tail of FBXL3 [93]. This is clearly the means by which SCF^{FBXL3} ubiquitin ligase recognizes CRY2 for degradation, as site-directed mutagenesis experiments altering the residues at the C-terminal tail of FBXL3 interfered with complexation [93]. In this same study, similar inhibition was seen when key residues in the CRY2 cofactor pocket were changed as well, suggesting that this interface between FBXL3

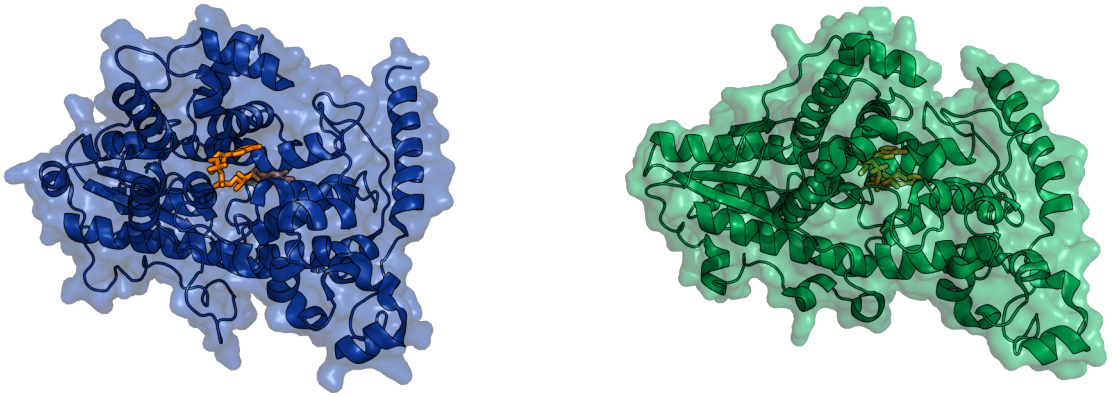


Figure 2.4: Crystal structures of FAD-bound mCRY2 PHR (left, PDB: 4I6G) and *Drosophila* photolyase (right, PDB: 3CVU). FAD is shown in orange in both structures. Note the open FAD binding pocket in mCRY2 whereas FAD is deeply buried in *Drosophila* photolyase [128].

and the FAD-binding pocket are crucial for molecular recognition. Further, experiments investigating the addition of FAD to preformed complex showed that FAD was capable of disrupting the complex, likely by competing for binding to the cofactor pocket.

2.4.3 Native Mass Spectrometry Analysis of CRY2–FBXL3–SKP1

The native mass spectrum of CRY2–FBXL3–SKP1 is shown in Figure 2.5a. Charge states +19–24 were observed with the +22 charge state displaying the greatest intensity. The peaks are narrow, revealing minimal adduction and heterogeneity. The mass was determined to be 127 340 Da, differing from the mass expected based on the sequence by 26 Da (204 ppm). No evidence for mass shifts due to phosphorylation or other posttranslational modifications was observed. Additionally, a 124 291 Da species was detected, which was later determined to be the complex with a truncated copy of FBXL3, FBXL3(Δ1–27). The identification of this mass is discussed in further detail in Section 3.4.2.

The complex was also subjected to collision induced dissociation (CID, see Section 1.1.3)

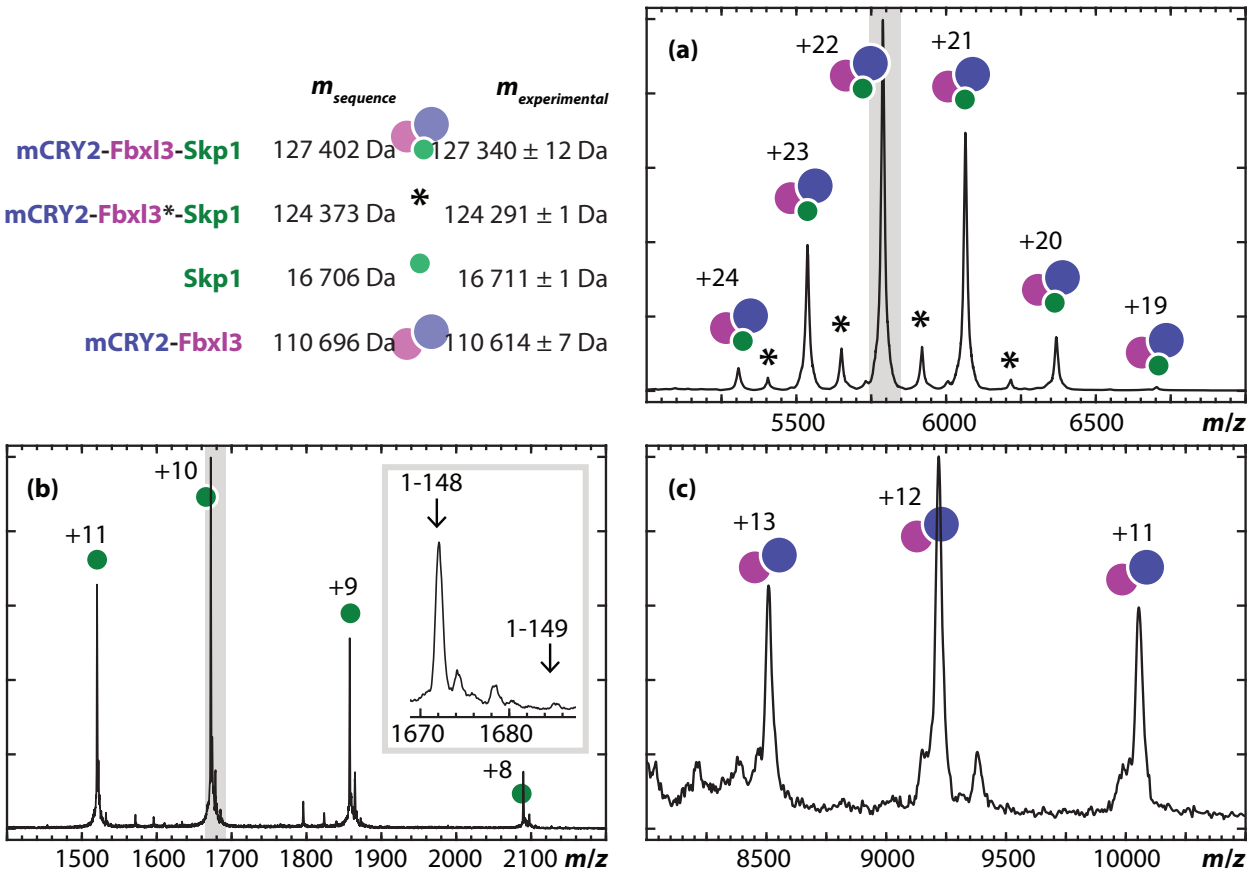


Figure 2.5: The native mass spectrum of the CRY2–FBXL3–SKP1 complex is shown in (**a**), with the identities, masses for the expected sequences (m_{seq}), and experimental masses (m_{obs}). The +22 charge state of the CRY2–FBXL3–SKP1 complex (grey region in (**a**)) was isolated using a quadrupole mass filter, and activated using collision-induced dissociation (CID). CID resulted in the formation of SKP1 and CRY2–FBXL3 product ions, show in (**b**) and (**c**), respectively. Note that the spectra shown are from separate experiments; conditions were adjusted to improve spectral quality for the product ion of interest. The spectrum of the SKP1 product ion shows that m_{obs} is less than the expected mass for the full sequence of SKP1 (amino acids 1–149, 16 834 Da), suggesting the removal of the C-terminal lysine (amino acids 1–148, 16 706 Da). The expected m/z values for these two forms of SKP1 are marked on the expansion of the experimental data for the $(\text{SKP1}+10\text{H})^{10+}$ ion. m_{obs} determined for the CRY2–FBXL3 product ion is consistent with the expected sequences without any phosphate group (+80 Da). Using this revised sequence for SKP1, m_{obs} for the CRY2–FBXL3–SKP1 complex is consistent with the revised m_{seq} and the absence of posttranslational modifications. A minor truncated form of the complex was also observed and is marked with (*). The location of the truncation was later identified as FBXL3(1–27), as described in Section 3.4.2.

to confirm the mass of individual subunits. SKP1 is ejected as the CID monomer product, appearing between 1500–2100 m/z with charge states +8–11 observed (Figure 2.5b). The observed mass of SKP1 revealed that the terminal lysine had been cleaved during expression/purification (Δm=123 Da, m_{Lys}=128 Da) and m_{seq} was adjusted accordingly for SKP1 and CRY2–FBXL3–SKP1. The CRY2–FBXL3 product was observed in the 8500–10250 m/z range with charge states +11–13 (Figure 2.5c).

2.5 Conclusions

A mammalian cryptochrome was characterized using native mass spectrometry. Native MS data show that mCRY2 has relatively weak interaction with FAD as compared to what would be expected from a plant or insect cryptochrome; subsequent K_d measurements and crystallization results agree well with these results. Additionally, it was observed that the mCRY2 PHR is more solution stable than the full length mCRY2. Additionally, the CRY2–FBXL3–SKP1 complex was observed as a heterotrimer by native MS, with no evidence for PTMs observed. Ultimately, crystal structures were obtained both of mCRY2 PHR binding to FAD and full-length mCRY2 in a complex with two SCF^{FBXL3} ubiquitin ligase proteins (FBXL3 and SKP1).

Chapter 3

ION MOBILITY MASS SPECTROMETRY REVEALS A CONFORMATIONAL SWITCH IN A MAMMALIAN CRYPTOCHROME – E3 UBIQUITIN LIGASE COMPLEX

3.1 Abstract

A ubiquitin E3 ligase – circadian clock protein complex CRY2–FBXL3–SKP1 has previously been crystallized and characterized. Here, native mass spectrometry experiments revealed a dynamic equilibrium of two conformational populations, one of which is previously uncharacterized. The nature of this equilibrium is probed through ligand interaction experiments to reveal a dependence on FAD. Ion mobility experiments are performed to garner Ω values of the proteins, and these values are matched with the known crystal structure of the complex. To determine an approximate model of the more extended conformer, the complex is analytically decomposed to determine its flexibility and bent at these sites to generate an ensemble of potential structures. Ω values of these structures are calculated and compared to the experimentally determined values of the extended conformer. Additionally, previously obtained cross-linking data is integrated, and a scoring function considering both these parameters is generated to eliminate all but the most promising structures.

3.2 Introduction

Cryptochromes (CRYs) are evolutionarily conserved flavoproteins that share a great deal of sequence homology with DNA photolysases and similarly are ubiquitous in both plants and animals [97]. In plants and insects, CRYs tightly bind the chromophore FAD and regulate circadian rhythm by acting as blue-light photoreceptors [98]. Mammalian CRYs have been found in every cell in the body, regulate circadian rhythm in a light-independent manner, and

are essential for maintaining a 24-hour clock in the absence of regular light/dark cycles [120]. Interest in cryptochromes and other circadian clock proteins has grown as disruptions in circadian rhythm have been linked to health issues including to type 2 diabetes, metabolic syndrome, gastrointestinal disorders, and certain types of cancer [100, 101, 130].

The body's circadian rhythm has a hierarchal structure in which the master clock is located in the suprachiasmatic nucleus in the hypothalamus [114, 117]. This clock controls sleep/wake cycles and is affected by exposure to light [113, 131]. Additionally, a cellular-level autonomous circadian clock functions independently of light/dark cycles and is driven by the heterodimer of the proteins CLOCK and BMAL1, which together activate the transcription of the *Cry* and *Per* genes [132, 133]. Both CRY1 and CRY2, products of the *Cry* gene, inhibit the CLOCK–BMAL1 heterodimer and consequently their own transcription, creating a negative feedback loop [118]. CRY2 is in turn marked for degradation by the SCF^{FBXL3} ubiquitin ligase complex, creating the periodicity that drives the clock [124, 134]. A mammalian CRY2 was recently crystallized both as a monomer and complexed to two members of the SCF^{FBXL3} ubiquitin ligase complex, FBXL3 and SKP1 [93]. Crystal structures of the CRY2 monomer reveal an open binding pocket and corresponding weak K_d for FAD binding [93]. The crystal structure and native mass spectrometry experiments of the CRY2–FBXL3–SKP1 complex revealed that the complex spontaneously assembles in a phosphorylation-independent manner, with the C-terminal tail of FBXL3 inserted into the FAD-binding pocket of CRY2 [93].

Native mass spectrometry (MS) is an increasingly important biophysical technique for studying dynamic systems for which X-ray crystallography and other solution-phase methods may provide limited information and require large amounts of sample [8, 135]. In native MS, gentle electrospray conditions and non-denaturing buffers are used, preserving noncovalent interactions and allowing proteins to retain native-like structures [14]. Native MS can provide information about subunit stoichiometry, heterogeneity, subunit arrangement, and ligand-binding [12, 88]. Compared with other biophysical techniques, mass spectrometry is rapid and sensitive, enabling much higher throughput when varying conditions.

When native MS is coupled with ion mobility (IM–MS) [42], further low-resolution structural information about multisubunit complexes can be obtained [7,86]. Specifically, IM–MS provides geometric information about analytes in the form of a collision cross section (Ω), the orientationally-averaged cross sectional area of a gas-phase macromolecule [58, 60, 136, 137]. These cross sections can be used as experimental restraints for generating models of subunit arrangement [86, 89, 138]. Interest has grown in using these integrative modeling approaches to predict structural arrangement of large, heterogeneous complexes, combining native IM–MS data with X-ray crystallography, homology modeling, electron microscopy, cross-linking, and molecular dynamics [78, 87, 90, 91, 139–142]. In general, these approaches have focused on generating interaction maps where the structures of individual monomers are thought to largely correspond to their crystal structure but the stoichiometry and subunit connectivity are not known. Conversely, there are also systems that have been described in which ion mobility has been used to characterize alternate conformers of a dynamic monomeric protein [143].

In this study, native mass spectrometry experiments show that CRY2–FBXL3–SKP1 exists as two distinct conformational populations that may also be present in solution. Ion mobility results show that the more compact conformer corresponds to the previously obtained crystal structure of CRY2–FBXL3–SKP1 [93], while the larger possesses a higher collision cross section and more extended geometry that has previously not been observed. These observations also shed light on recent cross-linking results that revealed a significant number of cross-links for CRY2–FBXL3–SKP1 that were not explained by the published crystal structure [94]. Herein we describe a novel modeling and scoring approach used to generate candidate structures of a previously unreported solution conformer and then evaluate them against ion mobility and cross-linking data.

3.3 Experimental Methods

All experiments were performed on a modified Waters Synapt G2 HDMS (Wilmstow, United Kingdom) where the traveling-wave ion mobility cell was replaced with a custom built RF-

confining drift cell that has been described elsewhere [144]. Nanoelectrospray (nESI) capillaries were pulled from borosilicate capillaries with an internal diameter of 0.78 mm to a tip with an orifice of approximately 1–3 μm using a micropipette puller (Sutter Instruments model P-97, Novato, CA). Electrospray voltage was supplied by a platinum wire delivering 0.6–1.0 kV to generate positively-charged ions. Sample solutions were typically 10 μM of protein in a 300 mM ammonium acetate buffer at pH 8.0. Ω values were determined by direct measurement of mobilities using the RF-confining drift cell. Data acquisition and instrument control were achieved using MassLynx 4.1. Charge states and analyte masses were assigned using in-house software that generates simulated mass spectra and uses a least squares minimization to fit these to experimental data (Appendix D).

Computational approaches were developed in house using the Python programming language, particularly the scientific libraries NumPy, SciPy, pandas, and matplotlib. The Gaussian network model was implemented using the spatial libraries and singular value decomposition (SVD) function in SciPy and NumPy [145,146]. Software to pivot structures, determine steric clashes, evaluate cross-link distances, and visualize results was also implemented in house using the aforementioned libraries. FORTRAN code for calculating the Ω values was graciously provided by Alex Shvartsburg [65] and recompiled in house as a Python extension [147] for direct accessibility by the main script. Visualization of candidate structures was achieved using PyMOL (Schrödinger, LLC), taking advantage of its Python API.

3.4 Results and Discussion

We have observed through native MS that the heterotrimer of CRY2–FBXL3–SKP1 occupies two conformational states, and Ω values determined from IM experiments reveal that the more compact of these conformers resembles the previously published crystal structure, while the nature of the larger conformer is more elusive. We present two hypotheses regarding the nature of the extended conformer. It is possible that, as the truncated complex does not occupy an extended state, the missing residues play a role in the extension, and reconstruction of these residues should yield a structure that agrees with the Ω measurements for the

extended conformer. However it is also possible that the structure itself is flexible and that a more global conformational change is responsible for the switch.

3.4.1 Native Ion Mobility Mass Spectrometry

Figure 3.1a shows the native MS of CRY2–FBXL3–SKP1 (127.3 kDa), which exhibits a bimodal charge-state distribution centered at +22 and +30. As charge state generally scales with solvent accessible surface area, it is somewhat dependent on the conformation of the protein [148, 149]. Native protein complexes are quite dynamic in solution but can become kinetically trapped upon entering the gas phase [150], effectively locking them into a relatively distinct conformation. Consequently, bimodal charge state distributions suggest two discrete populations of conformers with significantly different surface areas [143], with the conformer occupying higher charge states likely having a more extended structure.

The sample was analyzed using drift tube ion mobility, taken at a series of drift voltages as described in the Experimental Methods to enable the direct measurement of Ω values. Each distribution exists as a discrete population in drift time space (Figure 3.1b). Additionally, peak widths for the arrival time distributions are narrow and Gaussian in shape, which is typical of ordered structures and indicates that the experimental conditions are gentle enough to minimally affect the structure of the ions [151]. Drift times were measured at a series of drift voltages and used to calculate Ω values of each charge state, with charge states of the more compact distribution having a Ω values ranging from 63.0–65.0 nm^2 and those of the extended conformer ranging from 75.5–85.3 nm^2 (Figure 3.1c). The data show that the more highly-charged distribution occupies a more extended geometry than the less-charged distribution, confirming that the bimodal nature of the charge-state distribution is due to the presence of two discrete conformational populations.

For comparison, the collision cross section of the published crystal structure (PDB: 4I6J) was calculated using a linear combination of the projection approximation (PA) [63] and the exact hard-spheres scattering (EHSS) model [64, 65]. This linear combination has been described previously and has been shown to agree well with experimentally obtained values

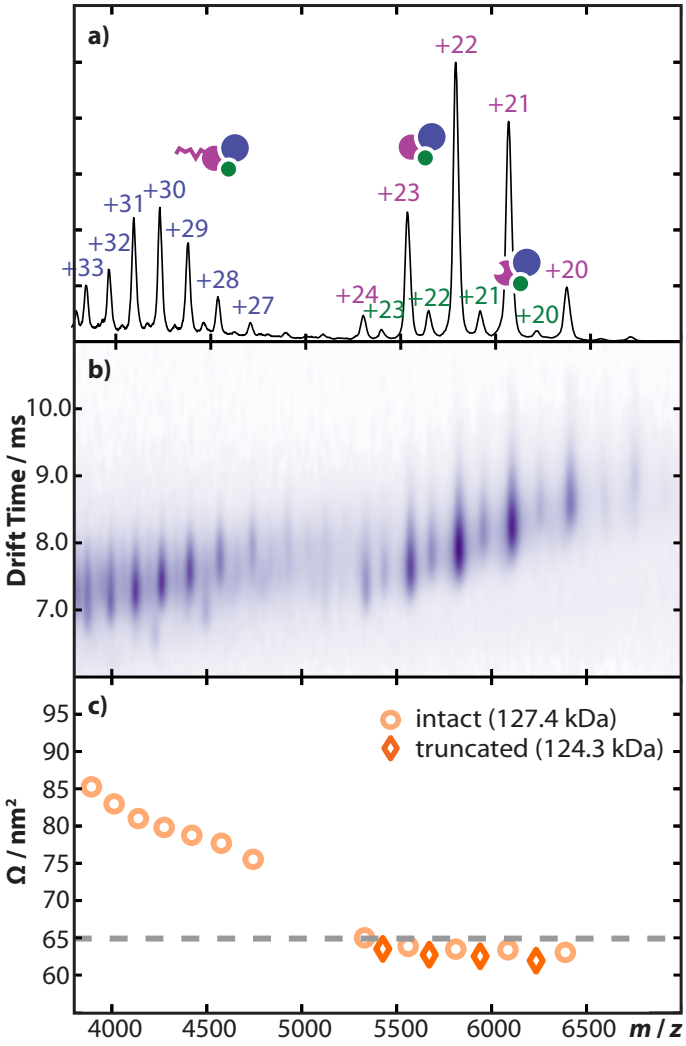


Figure 3.1: (**a**) Native mass spectrum of CRY2–FBXL3–SKP1. Two non-overlapping distributions of charge states suggest distinct populations of conformers. (**b**) IM–MS spectrum of CRY2–FBXL3–SKP1 with a drift voltage of 120 V. The extended and compact populations are clearly separated in drift time space. (**c**) Collision cross section values (Ω) of the compact and extended conformers (circles) and truncated complex (diamonds). The theoretical Ω was calculated from the published crystal structure (PDB: 4I6J, dashed line). Ω values of the truncated complex are comparable to the compact conformer and agree well with the calculated Ω, while the extended conformer ranges from 20–32% higher than the calculated Ω.

[72], and is significantly less computationally expensive than some alternatives. The linear combination Ω_{LC} used is as follows:

$$\Omega_{LC} = 0.84 * \Omega_{PA} + 0.22 * \Omega_{EHSS} \tag{3.1}$$

Ω_{LC} was determined to be 64.9 nm^2 (Figure 3.1c), which agrees well with the experimentally determined Ω values for the compact conformer. For comparison, Ω was also calculated using a scaled projection approximation [9] and determined to be 65.5 nm^2, which still compares favorably with the measured Ω values of compact conformer. Therefore the structure of the extended structure and the mechanism by which the switch occurs remain a mystery.

3.4.2 Factors Affecting Conformational Distributions in Solution

A 124.3 kDa species was also observed in the mass spectrum with a monomodal charge-state distribution more similar to the compact conformer of the 127.3 kDa species and a Ω value of 61.9–63.5 nm^2. The intensity of this lighter complex increased relative to the 127.3 kDa species when left at room temperature (data not shown), likely due to proteolytic cleavage of one subunit. There are no intermediate peaks between the two species to suggest terminal degradation. The identity of this missing mass was determined by native MS (Figure 3.2). To maximize the conversion to the lighter species and increase the abundance of the cleaved fragment, a sample of the protein complex was left at room temperature overnight. Native MS of the sample revealed near-complete conversion after overnight incubation. At low m/z, the fragment appeared as a 3+ ion, and the mass of the M+3H ion was determined to be 3048.5346 Da, which matches that of FBXL3(1–27) to an error of 0.4 ppm. The isotopic abundance of FBXL3(1–27) was calculated using IDCalc [152,153] and matched the observed intensities well (Figure 3.2).

Owing to there being no extended conformer for the truncated complex, the role of FBXL3(1–27) is clearly crucial. It has been observed previously that even small segments of protein protruding from otherwise globular structures can increase cross section significantly

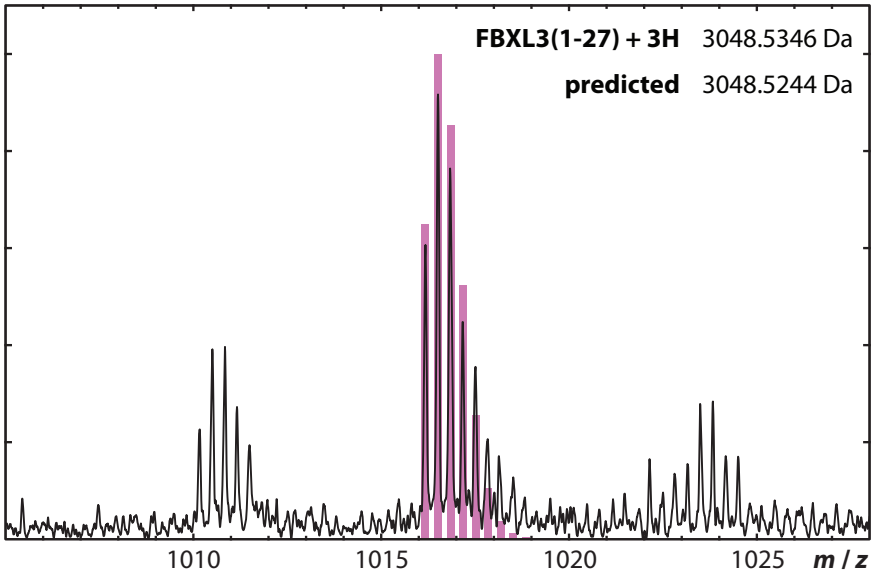


Figure 3.2: Native MS of the FBXL3(1–27) 3+ ion with predicted isotopic distribution. The mass spectrum of FBXL3(1–27) is shown, and is equivalent to the inset in the upper right corner of Figure 3.3. The predicted isotopic distribution was calculated from the FBXL3(1–27) sequence using IDCalc, and is shown in pink. The series of peaks at ~1010 m/z correspond to the loss of a water, and the peaks at ~1023 m/z correspond to a single sodium adduct.

[143]. It is interesting to observe that FBXL3(1–34) do not appear in the published crystal structure of the complex [93], suggesting some degree of flexibility but leaving the structure of the N-terminus a mystery. It is important to note that the protein sample we analyzed differed in sequence compared to the published crystal structure, with an additional serine and histidine at the N-terminus, and therefore any further analysis will be considering 36 missing residues. In modeling this structure, it was assumed that the extending loop would be flexible but ordered to some degree. Two extreme cases were considered: a linear chain (φ = -180°, ψ = -180°) and a tight α-helix (φ = -57°, ψ = -47°), with the actual secondary structure of the fragment likely being somewhere in between. To model the effect of the N-terminal residues of FBXL3 extending from the compact conformer, the missing residues (1–36) from the FBXL3 terminus were reconstructed in PyMOL. Proline residues were replaced with alanine residues to simplify the conformational space and dihedral angles were set to either

a linear chain or α-helix. The angle of the extension relative to the complex was varied to account for the rigidity of the model when calculating Ω values. For each structure, 60 different angles were used to position the extended terminus relative to the remainder of the structure, and whichever angle resulted in the largest Ω value was considered as the most probable gas-phase structure considering Coulombic repulsion. The largest structures for each of these models have cross sections of 70.8 nm^2 for the α-helical model and 75.5 nm^2 for the linear chain model. The linear model was comparable to the smallest of the extended conformers, however neither value compared with those for the extended conformer at higher charge states.

Additionally, we modeled the complex as if the N-terminus were unraveling by deleting N-terminal residues that were in the original crystal structure and reconstructing them with either the previously described linear or α-helical geometry. Reconstruction of the complex in this manner resulted in large increases in calculated Ω_{calc} (Figure 3.3b). The linear model was, as expected, much larger in cross section compared to the α-helical model. Lower charge states of the extended conformer fit well within the bounds of these two models as the extent of unraveling increased. Experimentally determined Ω values for the higher charge states of the extended conformer were significantly larger than those calculated unless a considerable amount of unraveling was used in the model. Regardless of the increase in size, this modeling approach was intended to produce two extreme cases with which to bracket the data. Thus, as the linear model likely does not represent the gas-phase structure we have to imagine that the actual size contribution of the extended terminus is much smaller. Further, the amount of unraveling needed to obtain large cross sections is unrealistic. While it possible that the extended N-terminus of FBXL3 makes some contributions to the extended conformer, it is not a large enough increase to account for the experimentally observed cross sections.

Additionally, native mass spectra were acquired of the complex with 100 μM FAD added to solution. No subunit dissociation was observed after incubation with FAD for one hour (Figure 3.4b) — although previously published competition experiments had shown that FAD in sufficient concentration disrupts the interaction between CRY2 and FBXL3 [93]

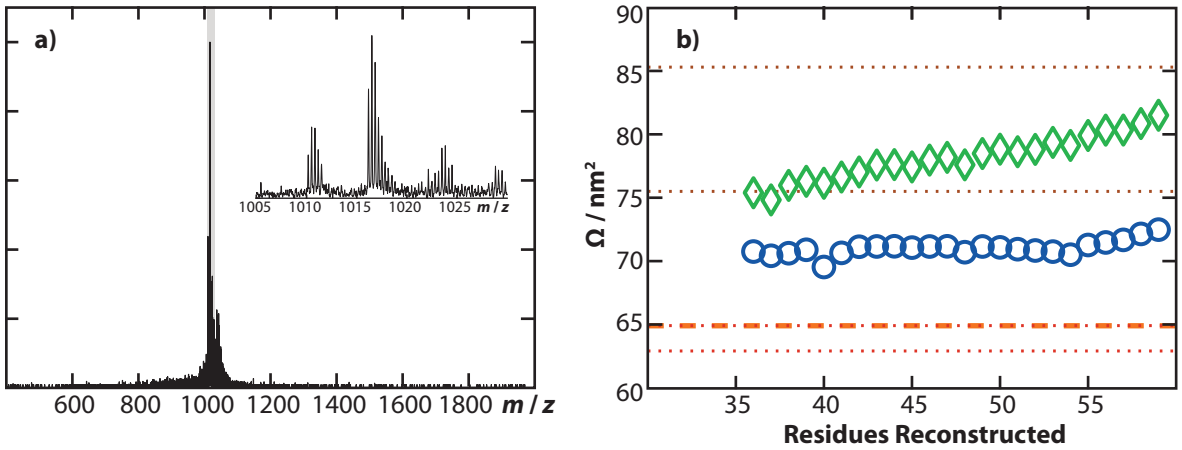


Figure 3.3: (**a**) The mass spectrum of the complex incubated overnight reveals that is little intensity at low m/z, with the exception being the fragment truncated from the main complex, which was identified as FBXL3(1–27). The region containing the fragment is highlighted with a gray box and is shown enlarged in the upper right corner. (**b**) Residues 1–36 from the N-terminus of FBXL3 are reconstructed *in silico* as either an α-helix (blue circles) or linear chain (green diamonds). The angle that the segment protrudes from the complex was varied to obtain the maximum Ω_{calc}, which would be expected for a partially disordered, extending structure due to Coulombic repulsion. Reconstruction of the missing residues leads to an increase of 9% for the α-helical model and 16% for the linear model. Larger structures can be accessed by modeling the unraveling of the N-terminus from the rest of the complex by deleting additional residues and then reconstructing them. For comparison, the range of Ω values for the compact conformer is shown with a dotted red line, the range of Ω values for the extended conformer is shown with a dotted rust-colored line, and Ω_{calc} for the unmodified crystal structure is shown by a dashed orange line.

— however the extended conformer is no longer visible in the native MS and a mass shift corresponding to the mass of FAD is observed. The relative intensities reflect an approximate K_d of 296 μM, though FAD competes with the C-terminal tail of FBXL3 for binding [93]. Upon removal of FAD by buffer exchange, the extended conformer is once again visible (Figure 3.4c). It should be noted that the relative intensities of the extended conformer differ from those in Figure 3.1a, as these experiments were carried out on different days.

It is clear from the reappearance of the extended conformer after removal of FAD that the two conformers are in equilibrium. It was not possible to monitor these kinetics because

of the time required for effective buffer exchange, however the return of the extended conformer to comparable levels before buffer exchange supports an equilibrium of conformers. Concentrations of FAD at concentrations as low as 100 μM are able to shift the equilibrium to a compact geometry, even without complete binding. It is likely there is a transient interaction with FAD that helps the complex assume a compact state and that the effects of

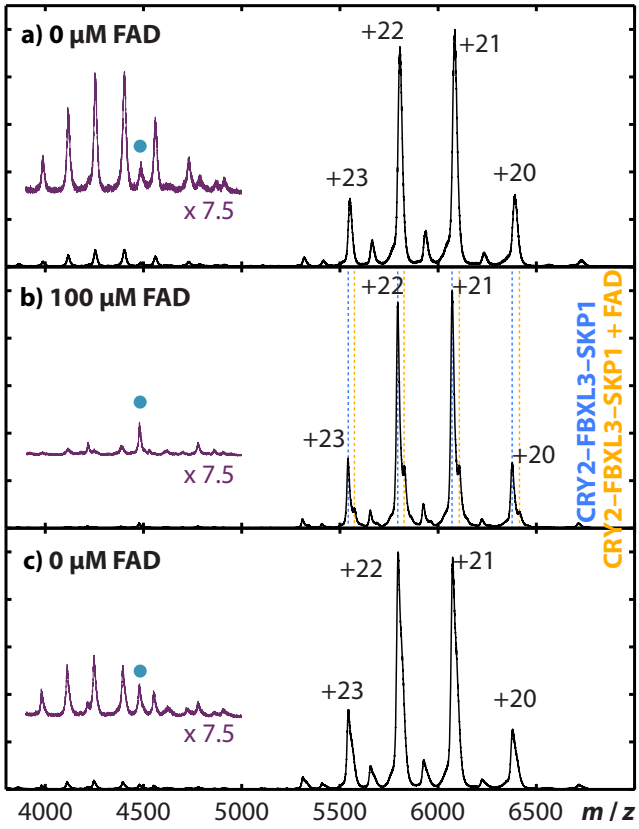


Figure 3.4: (**a**) CRY2–FBXL3–SKP1 is electrosprayed in a 300 mM ammonium acetate buffer at pH 8.0 with no FAD added (extended conformer magnified in upper left corner). The blue circle indicates a chemical impurity that acts as an internal standard for determining the relative amount of extended conformer. (**b**) The sample is electrosprayed in a 300 mM AmAc buffer at pH 8.0 with 100 μM FAD. Adduction of FAD to the compact conformer is apparent. The extended conformer signal is significantly diminished. (**c**) The complex is buffer exchanged from a buffer containing 100 μM FAD to one with no FAD. Upon removal of FAD, the extended conformer is again visible.

the interaction persist for some time if the equilibrium is totally shifted without complete binding. Unfortunately, quantifying this equilibrium from the relative abundance of each conformer is difficult due to the likelihood of non-uniform response factors between the two conformers [154].

3.5 Computational Results

Native MS data revealed that CRY2–FBXL3–SKP1 appears as a bimodal charge-state distribution with centers at +30 and +22. Our results showed that upon addition of FAD, the higher charge state distribution is no longer seen, however upon removal of the FAD by buffer exchange it reappears. Additionally, a truncated species of 124.3 kDa is seen with a charge-state distribution similar to that of the lower charged distribution. The absence of an extended form of the truncated complex is intriguing, and suggests that these missing residues may play a role in the conformational switch. If the N-terminus of FBXL3 were to extend from the complex, it would be vulnerable to proteolysis in solution and also be able to accommodate additional charges during ionization, further encouraging the possibility it could play a role in the conformational switch. That the ratio of truncated to intact complex increases as the solution is left at room temperature supports that either some residual protease is present in the sample or that an autocleavage is occurring, though there is no evidence to favor either possibility.

3.5.1 Comparison with Solution Cross-linking

It is also worth noting that previously obtained cross-linking data by Hoopmann et al. revealed a new cross-linking methodology, Kojak, found cross-links in CRY2–FBXL3–SKP1 that did not agree with the published crystal structure [94]. The authors set a distance cutoff of 30.0 Å for all cross-links given the spacer arm length of the reagent and the presumed flexibility of the protein complex, yet 26% of the identified cross-links did not fall within this cutoff. Conversely, the Cop9 signalsome and the *S. pombe* 26S proteasome were also analyzed using Kojak, but no cross-links exceeding the 30.0 Å distance cutoff were reported.

Although these cross-links in CRY2–FBXL3–SKP1 were attributed to potential cross-linking with other complexes in solution, it seems prudent to consider that distant cross-links are actually due to the alternate conformers we have observed herein.

3.5.2 Computational Results for Global Conformational Changes

Large changes in conformation resulting from conformational flexibility of the complex were also considered for potential increases in cross section and validation of cross-links that were not within a 30.0 Å cutoff described in Hoopmann et al. [94]. We sought to develop an approach to identify flexible regions of the structure and then bend the structure to access alternate conformations. First, flexible points in each subunit were identified in an automatic fashion using a Gaussian network model (GNM) [155]. This algorithm was originally proposed for use on monomeric proteins, and the authors provided no examples of it being validated for multiprotein complexes. The algorithm was implemented using Python as described in Section 3.3. Briefly, the protein is approximated as a graph, with α-carbons representing the nodes and adjacency determined by the distance between the α-carbons, with the critical radius defining adjacency set to 10.0 Å. The adjacency and valency of the nodes in a graph can be represented by a Kirchoff matrix, which is then decomposed using singular value decomposition. The eigenvector corresponding to the lowest non-zero eigenvalue is called the Fiedler vector, and intersections of the Fiedler vector represent points of flexibility in the protein structure. This analysis was performed on each subunit individually, and it was found that each subunit had a single pivot point. Therefore three flexible points were identified in this manner, with a fourth in FBXL3 determined by visual inspection (Figure 3.5), which will be referred to as FBXL3*. This analysis is discussed in further detail in Appendix A and includes the full Python code used.

The structure was bent at this point over a range of angles in a manner accessing the entire conformational space. Due to the asymmetry of the structure, it is only possible to access the complete conformational space by rotating about two orthogonal axes by 2π radians and rotating about a third by π radians. For each new structure generated, the collision cross

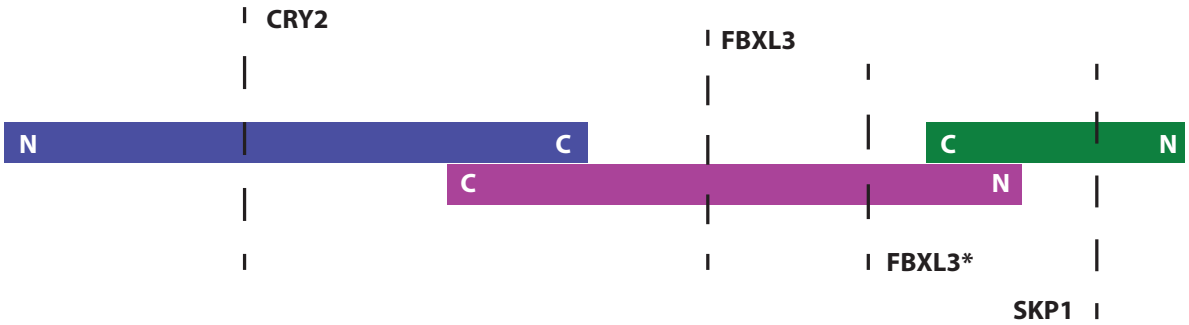


Figure 3.5: Diagram of decomposed domains. Bars do not represent actual relative protein sizes and locations of the pivot points are approximate. One pivot point was identified per protein via the GNM, and these are labeled as the subunit at which they are located. One additional pivot point was identified visually (FBXL3*).

section was calculated and the distance between pairs of cross-links is calculated. Increases in Ω_{calc} and decreases in the distances between cross-linked residues were favored. Additionally, steric clashes that occur from the rotations were determined for each structure and highly clashing structures were disfavored in the model. These three metrics were integrated into a scoring function of three terms as follows

$$S = \left(\frac{\Omega_{calc}}{\langle\Omega_{exp}\rangle}\right) \times \left(\sum \Delta \frac{1}{1 + e^{0.5(z_i - \mu)}}\right) \times \left(\frac{n_{atoms} - n_{clashing}}{n_{atoms}}\right) \tag{3.2}$$

Briefly, the first term scores the calculated Ω values as a ratio of the structure's Ω_{calc} to the average of the experimental extended conformer collision cross section values $\langle\Omega_{exp}\rangle$. Ω_{calc} values of each structure were calculated using a linear combination of the PA and EHSS as described above. To ensure strong statistical confidence, for each structure 16 different values were calculated, such that the uncertainty, as defined by the 95% confidence interval, would be less than 3.0% for any given structure. The max error among all values in the dataset was 2.1%, well within the 3.0% tolerance for drift tube measurements. The largest increases in Ω were seen for the FBXL3 rotation point, with the largest structure generated having a cross section of 73.6 nm^2, a 13.4% increase in CCS compared with the value calculated for

the PDB structure.

The second term scores how well a structure satisfies the hard cutoff that was used to evaluate cross-links in Hoopmann et al. [94] Rather than using a hard cutoff of 30.0 Å, however, we chose a sigmoidal function centered at 30.0 Å ($\mu = 30.0$). As there are two conformational populations, it is not necessary for any given structure to satisfy all the cross-links. Rather, a structure should be scored well if the distance between a pair of cross-linked residues is reduced compared to the published crystal structure. This was evaluated by calculating the score for a cross-link in the original structure and the score for that corresponding cross-link in the rotated structure. If the score improves, then the difference is in score is calculated. These differences are then summed to determine the middle term for the score of that cross-link. In doing so, cross-links that do not improve in the rotated structures do not penalize the score. Further, by centering the sigmoidal function at 30.0 Å, changes in distance from above the original cutoff to below are weighted the most heavily.

The third term considers the steric clashing of the generated structure. As every structure is generated by first dividing the structure into two parts at the mobile hinge point and then rotating one of those parts, the structures can be said to have a rotating half and a static half. Briefly, a Euclidean distance matrix (D) is constructed where, for every atom in the rotating half of the complex, the distance is computed between it and every atom in the static half. Additionally, a minimum distance matrix (T) of the same shape is constructed where for every pair of potentially clashing atoms the sum of their two van der Waals radii is computed. The minimum distance matrix is subtracted from the Euclidean distance matrix, and negative values indicate a pair of atoms whose distance is less than the sum of their van der Waals radii. This pair of atoms is considered to be sterically clashing. The fraction of atoms that experience no clashes in the structure becomes the third term of the scoring function.

The values from the scoring function are shown in Figure 3.6, comparing the contributions of each term to the calculated score. The CRY2 and FBXL3 pivot points show an overall much higher degree of correlation and higher total score than either SKP1 or FBXL3*. Struc-

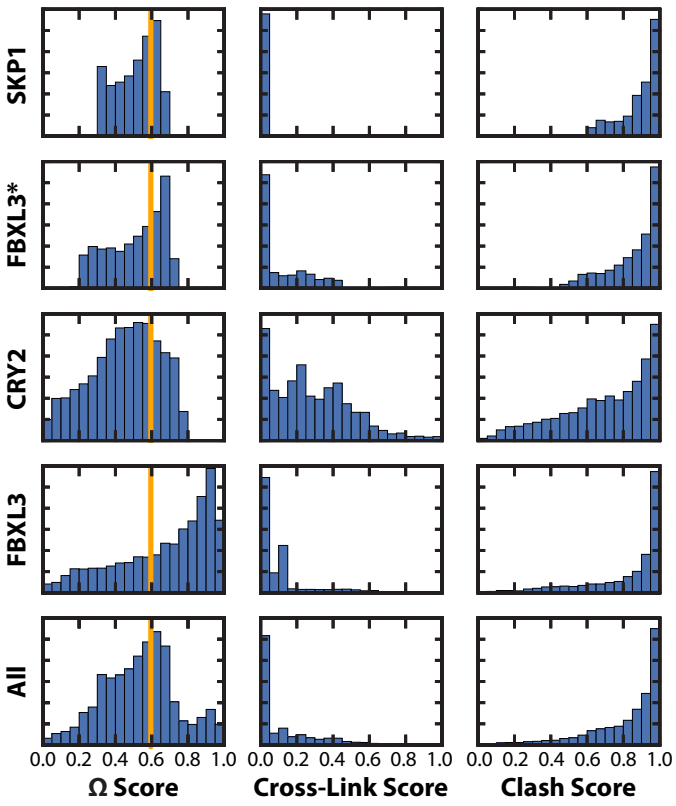


Figure 3.6: Histograms of structures and their scores are shown, broken down by score term and by pivot point, with the bottom row the distributions for each term for all structures. For the Ω score term, an orange line denotes the score for the original structure. It is notable that the majority of structures experience a decrease in Ω_{calc} for the SKP1, FBXL3*, and CRY2 pivot points, though there are structures with increased Ω for each of them. The FBXL3 pivot point shows the greatest increases in Ω_{calc}, whereas the CRY2 pivot point overwhelmingly shows the best improvements in cross-link scores.

tures generated from the CRY2 pivot point scored significantly higher due to contributions from improvements in cross-link distances, while structures from the FBXL3 pivot point showed the greatest increases in Ω. Figure 3.7 shows the correlation between scores for the clash term and both the Ω term (Figure 3.7a) and cross-link term (Figure 3.7b). Ω scores and clash scores have a weak but positive correlation ($r = 0.80$), which is to be expected as structures that have a high degree of clashing will also have more compact geometries. There is little to no correlation ($r = -0.53$) for the clash and cross-link scores. The linear regression

does yield a negative slope, which is due to the cluster of structures with low cross-link but high clash scores. This quite explicitly justifies the need for a clash term, as structures that are allowed to occupy impossible geometries would have an advantage bringing cross-linked residues closer than is physically possible. This is perhaps best illustrated by the structure in the model with the highest cross-link score having 1095 clashing atoms (out of 8279 total atoms).

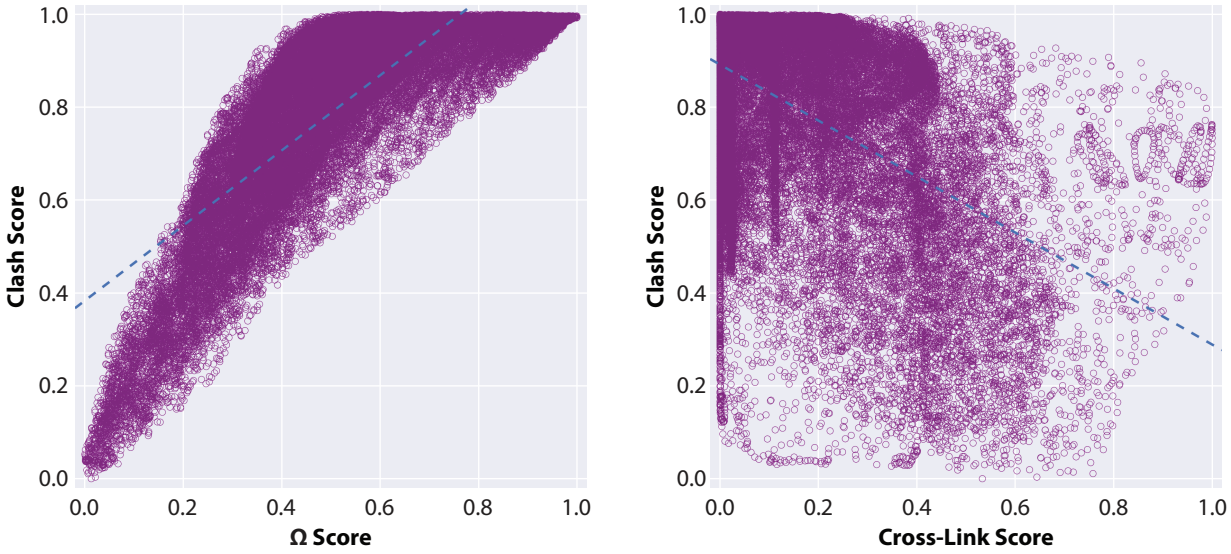


Figure 3.7: As expected, there is a moderate correlation between the clash term and Ω term, as structures with a high number of clashes will have a smaller size. No correlation is seen for the cross-link and clash term, however note the high density of structures with high clash scores and low cross-link scores. Giving the protein freedom to access overlapping conformations leads to decreases in cross-link distances that are not physically possible.

It should be noted that due to contributions from the cross-linking term, the highest scoring structures for the FBXL3 pivot point are surprisingly not those that showed the largest increases in Ω (Figure 3.8a). The highest scoring of all structures came from the CRY2 pivot point (Figure 3.8b), with score of 0.487 and a Ω of 65.4 nm^2. This is less than a 1% increase in Ω compared to the crystal structure, however it is important to note that while this is not likely the structure of the extended conformer, it verifies the flexibility of the

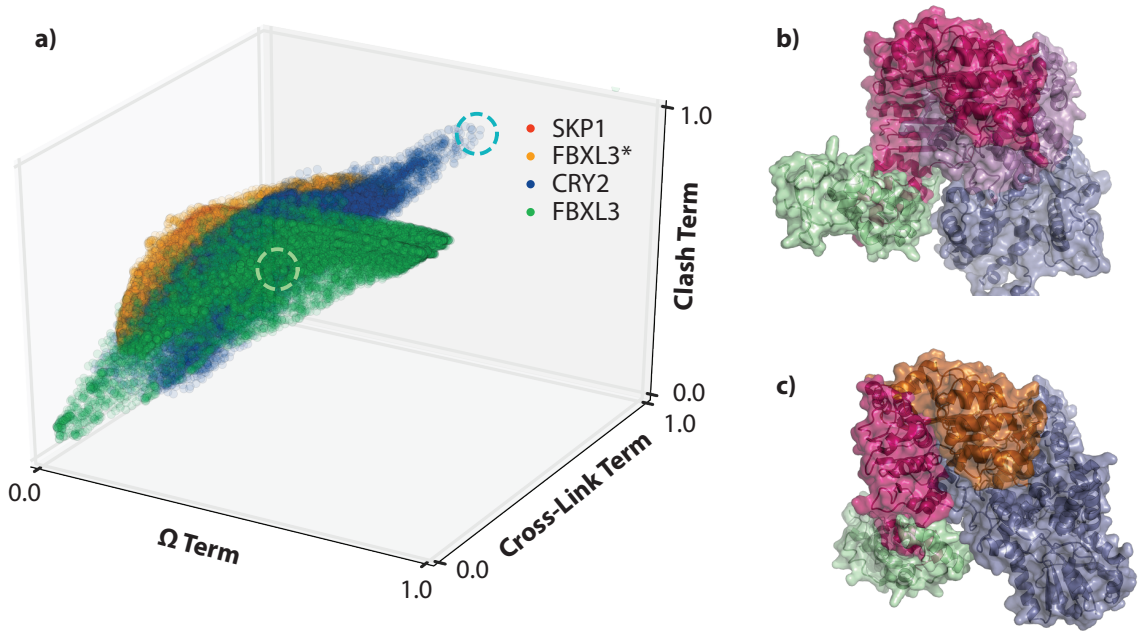


Figure 3.8: (**a**) The contribution of each score term to the final score is shown for each domain. The top scoring structures for the CRY2 pivot point (blue dashed circle) showed the greatest improvements in cross-link distances compared to other pivot points. The greatest increases in Ω_{calc} came from the FBXL3 pivot point, however the top scoring structures from this group (green dashed circle) experienced more modest Ω_{calc} gains with the cross-link distances contributing more to the total score. Crystal structures of the top scoring structure from (**b**) the CRY2 pivot point (blue/purple) and (**c**) the FBXL3 pivot point (pink/gold) are shown with alternate coloring indicating the interface between the mobile and static halves of the complex.

CRY2 pivot point identified by the GNM, as transient conformational changes in solution can facilitate cross-links. Thus, intermediates and transition states for the conformational switch are likely the source of the cross-links described by Hoopman et al. that did not match up with the crystal structure. While the FBXL3 pivot point does not score as highly for improving cross-links, it has the highest contributions to Ω and should therefore be noted as being significant. The structure of the highest scoring structure from the FBXL3 pivot point is shown in Figure 3.8c.

3.6 Conclusions

We identified a conformational switch occurring for a mammalian cryptochrome – E3 ligase complex. Ion mobility mass spectrometry has revealed that there are two conformers present, a more compact conformer that resembles the previously characterized structure and a more extended conformer. Native MS data has shown that there is a truncated form of the complex that cannot undergo the switch and also that the presence of FAD shifts the conformational equilibrium to the more compact form.

Two modeling approaches were considered. First, the N-terminal residues of FBXL3 were reconstructed using either a linear or α-helical geometry and Ω values were calculated of the models. When these Ω_{calc} did not account for the greater size of the extended conformer, we simulated the unraveling of the N-terminus to obtain larger values. A significant amount of unraveling was required to obtain these higher values, indicating that an extension of the N-terminal tail was not enough to account for the size difference. Alternatively, more global conformational changes were considered. Pivot points were identified on each subunit and the complex was bent at these locations, generating an ensemble of structures. For each structure, the Ω value was calculated, cross-linking distances were evaluated, and steric clashing was considered. Structures were ranked using a scoring function of these three parameters.

Our scoring function revealed several promising candidates from two of the pivot points, FBXL3 and CRY2. None of these structures were large enough to match the ion mobility data, however cross-linking scores strongly suggest that the CRY2 subunit is quite flexible. Meanwhile, the FBXL3 pivot point unlocks some of the physically largest structures in the ensemble. It should also be noted that although our automatic domain decomposition analysis revealed three pivot points, we were able to visually identify a fourth, raising the possibility that there are other potential pivot points. The model also operated on the published crystal structure, without considering the reconstructed residues from the FBXL3 N-terminus. Additionally, while we treated the two pivoting domains of the protein as being

rigid for computational frugality, in reality they must have some degree of flexibility, and the interface between domains may behave more like a tether than a hinge. Ultimately, it is likely some combination of multiple pivot points, unidentified flexibility, and contributions from the FBXL3 N-terminus that account for the conformational switch.

We present a novel modeling approach to characterizing protein complex flexibility using gas phase data as restraints. Many existing modeling platforms exist that focus on integrating high resolution subunit structures with larger interaction maps, and the modeling approach used herein incorporates and builds upon approaches described previously, namely automatic domain decomposition and use of a scoring function [138]. Our model is quite rapid to implement, and while we withheld molecular dynamics simulations in order to focus on generating a large ensemble of structures with minimal computational expense, we feel that our model complements these approaches well. Further investigation is certainly warranted to characterize the CRY2–FBXL3–SKP1 conformational switch and gain further insight into the mechanism of F-box recognition.

Chapter 4

NATIVE MASS SPECTROMETRY FOR THE RAPID CHARACTERIZATION OF THE ASSEMBLY OF FUSION PROTEINS FOR STRUCTURAL BIOLOGY

Portions of the work presented in this chapter have been published previously in

Connie Lu, Stewart Turley, Samuel T. Marionni, Young-Jun Park, Kelly K. Lee, Marcella Patrick, Ripal Shah, Maria Sandkvist, Matthew F. Bush, and Wim G.J. Hol. Hexamers of the type II secretion ATPase GspE from *Vibrio cholerae* with increased ATPase activity. *Structure*, 21(9):1707–1717, **2013**.

4.1 Introduction

The type II secretion system (T2SS) is a large protein apparatus that Gram-negative bacteria use to secrete fully folded protein complexes from the periplasm into extracellular space [156]. Bacteria such as *V. cholerae*, *P. aeruginosa*, and *L. pneumophilia*, to name a few, secrete their pathogenic toxins via the T2SS. The T2SS is therefore a clinically significant target for novel approaches to treating these infections. Briefly, the T2SS is composed of four subassemblies: the inner-membrane platform, the outer-membrane platform, the pseudopilus, and the cytosolic ATPase. It is hypothesized that secretion occurs as the ATPase $GspE^{EpsE}$ drives the assembly of the pseudopilus, which mechanically pushes the fully assembled protein complex product into extracellular space through an outer membrane pore.

The T2SS is a dynamic structure composed of both membrane-bound and soluble proteins. Subcomplexes, including the cytosolic ATPase $GspE^{EpsE}$, do not form structures resembling those found in nature in the absence of interactions with neighboring subunits. $GspE^{EpsE}$ is known to be a soluble protein but interacts with the membrane-bound inner-platform protein $GspL^{EpsL}$ via its N1 domain [157, 158]. A version of $GspE^{EpsE}$ with its

N1 domain truncated ($^{\Delta N1}GspE^{EpsE}$) has been crystallized in isolation [159], but size exclusion chromatography (SEC) has shown that $^{\Delta N1}GspE^{EpsE}$ is predominantly monomeric in solution and exhibits very weak ATPase activity [160]. It is interesting to note, however, that SEC fractions of $^{\Delta N1}GspE^{EpsE}$ corresponding to higher-order oligomers had greater ATPase activity (but exhibit very low absorbance at 280 nm), suggesting that oligomerization is necessary for $GspE^{EpsE}$ to be biologically active. Further, it was hypothesized that the native oligomeric state of $GspE^{EpsE}$ is a hexamer based on site-directed mutagenesis [161], and many homologs of $GspE^{EpsE}$ from other systems are hexameric [162–165]. To enable oligomerization, $^{\Delta N1}GspE^{EpsE}$ was fused via an amino acid linker to Hcp1 (Figure 4.1), a virulence protein from *P. aeruginosa* that forms a hexamer [166].

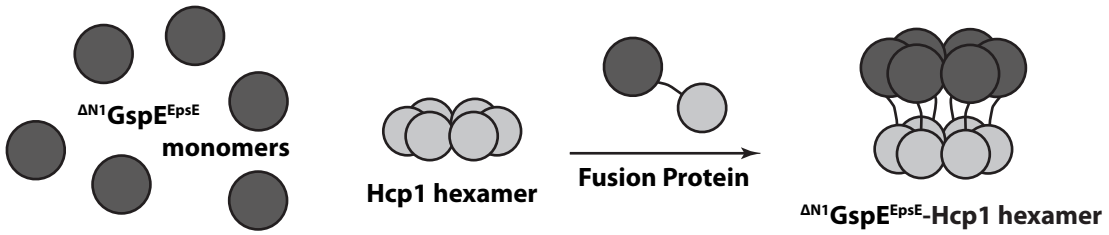


Figure 4.1: $^{\Delta N1}GspE^{EpsE}$ is monomeric in the absence of neighboring T2SS subunits [160]. To assist with oligomerization it was fused to Hcp1, a protein known to form a hexamer in isolation. Subsequent native MS experiments served to validate the stoichiometry of the fusion protein complex.

Recently native mass spectrometry was used to rapidly characterize fusion proteins, which were engineered to enable the structural and biochemical characterization of otherwise difficult complexes from the type II secretion system (T2SS) [167]. Mass spectrometry is an increasingly useful technique for structural biology. Once limited to small molecules, the advent of soft ionization methods such as matrix-assistance laser desorption ionization (MALDI) and electrospray ionization (ESI) has enabled the analysis of larger biomolecules, in particular proteins. Conventional protein mass spectrometry has involved the analysis of peptides and denatured intact proteins to obtain information about sequence and post-translational

modifications. Interest has grown in using nanoelectrospray ionization (nESI) of nondenaturing solutions, such as aqueous ammonium acetate buffers at biologically relevant ionic strength and pH, to transport fully folded protein complexes into the gas phase with minimal disruption to their native conformation and noncovalent interactions [8–13, 19, 135, 168]. Native mass spectrometry is used to obtain complementary structural information such as stoichiometry and protein-ligand interactions from the analysis of these intact, native-like protein complexes. Combined with subunit masses determined using conventional mass spectrometry experiments, stoichiometry of subunits can be determined from the observed mass of intact protein complexes.

The relationship between protein sequence and structure is complex, and fusion proteins will not necessarily form the targeted structures. A fusion protein is expressed as a single peptide chain, and each terminus must fold into the corresponding original fusion partner, ideally with minimal structural contribution from the linker. Composition and length of the

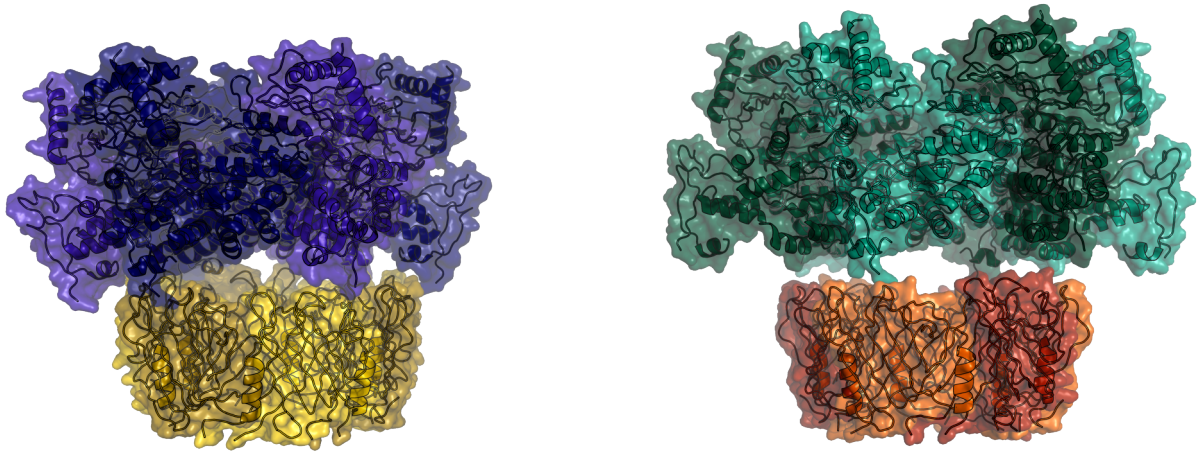


Figure 4.2: Crystal structures of hexameric $^{\Delta N1}GspE^{EpsE}$–6aa(GSGSGS)–Hcp1 (PDB: 4KSS), with the quasi C_6 ATPase shown in purple and Hcp1 in yellow (left), and hexameric $^{\Delta N1}GspE^{EpsE}$–8aa(KLASGAGH)–Hcp1 (PDB: 4KSR), with the C_2 ATPase shown in teal and Hcp1 in orange (right).

amino acid linker directly affect the folding of the fusion protein, with misfolding leading to aggregation, binding of heat shock proteins and molecular chaperones, or biologically irrelevant structures. Once an expressed fusion protein is purified, techniques including SEC and native gels are used to determine if a fusion protein is stable and forms oligomers. While these techniques are rapid (typically ~1 hour) and low cost, they often do not possess sufficient resolution to distinguish between oligomeric states. Native mass spectrometry is rapid, high resolution, and requires mere micrograms of sample. Herein we describe the role native mass spectrometry played in rapidly characterizing $^{\Delta N1}GspE^{EpsE}$ fusion proteins and present a narrative of the optimization challenges to engineer a fusion protein that was amenable to subsequent crystallization. Four fusion protein complexes were crystallized in total: $^{\Delta N1}GspE^{EpsE}$–5aa–Hcp1 and $^{\Delta N1}GspE^{EpsE}$–6aa–Hcp1 (PDB: 4KSS) with quasi C_6 symmetry, and $^{\Delta N1}GspE^{EpsE}$–7aa–Hcp1 and $^{\Delta N1}GspE^{EpsE}$–8aa–Hcp1 (PDB: 4KSR) with C_2 symmetry [167].

4.2 Material and Methods

4.2.1 Materials

Effective mass spectrometry analysis requires that analytes are free of non-specific adducts in order to produce clean, resolvable peaks that reflect the accurate mass of the analyte. Adduction of ions such as sodium, which is virtually ubiquitous in standard protein storage buffers, can broaden and shift peaks to a higher m/z [169]. Even minimal adduction may lead to biased mass assignments, while more extensive contamination will broaden peaks until they are no longer resolvable. Native mass spectrometry, which uses biologically relevant ionic strengths in order to preserve quaternary structure, is possible through the use of ammonium acetate as both a buffer and source of ionic strength. At concentrations mimicking the original sodium chloride concentration, ammonium acetate has been shown to be able to successfully maintain protein complex structure in many cases. Cofactors and metal ions that are necessary for structure and function are often a part of an ammonium acetate buffer

in a native mass spectrometry experiment.

$^{\Delta N1}GspE^{EpsE}$ fusion proteins were stored in a buffer of 20 mM Tris, 500 mM NaCl, 1 mM TCEP, and 5% glycerol at pH 8. Proteins were buffer exchanged using a Corning Spin-X UF centrifugal concentrator with a 10 kDa molecular weight cutoff (MWCO) into 500 mM ammonium acetate adjusted to pH 8 with ammonium hydroxide. Unless otherwise indicated, the ammonium acetate buffer also included 50 μM ADP and 50 μM $MgCl_2$. The concentrators were initially washed two times with 500 μL of 70% isopropanol, followed by two washes with the ammonium acetate buffer. Protein was added to 500 μL of the buffer and exchanged four times by centrifuging at 12000 x g for 20 minutes or until volume decreased to ~25 μL. The amount of protein initially added was adjusted so that the final concentration of protein after centrifugation would be ~10 μM.

4.2.2 Native Mass Spectrometry

nESI is conducted using a standard Waters Z-spray source nESI on a translatable stage. nESI tips are pulled from borosilicate capillaries (0.78 mm I.D.) using a micropipette puller (Sutter Instruments model P-87, Novato, CA) to yield a tip with an orifice between 1–10 μm. As little as 3 μL of sample in an aqueous buffer is added to the capillary, with up to 8 μL being added for solutions with high organic content. A platinum wire electrode (0.127 mm I.D.) is inserted into the open end of the capillary until it makes contact with and supplies a voltage to the solution. While gold plating the capillary is the most common method of delivering voltage [18], a platinum wire was selected because it makes direct electrical contact with the sample, is reusable, and lowers both material costs and labor. Further, whereas nESI often employs a voltage of 1.0–1.5 kV, we have been able to obtain high quality signal with a typical voltage of 0.6–1.0 kV for aqueous samples. No backing gas is required as the flow of solution is electrostatically driven. This results in the generation of smaller droplets [31], which are favorable in nESI because they reduce adduction to the protein and enhance desolvation.

4.2.3 Charge Reduction

Cation to Anion Proton Transfer Reaction (CAPTR) spectra are acquired using a previously described implementation [170, 171]. Briefly, perfluoro-1,3-dimethylcyclohexane (PDCH) vapors are passed through a glow discharge ionization source to produce $[PDCH\text{-}F]^-$ anions, which are quadrupole filtered and accumulated in the trap cell on the instrument [172]. Native-like protein complexes are then transmitted though the stored anions to initiate proton transfer reactions. Both unreacted precursor cations and charge-reduced cations are then mass analyzed.

4.2.4 Data Analysis

RAW files were loaded in MassLynx, integrated, calibrated, and exported as a two column text file. The spectra were analyzed using the in-house developed software package NativeFit (Appendix D). The software consists of a set of functions written in the Python programming language that makes use of the many scientific and data analysis libraries within the Python ecosystem, specifically NumPy, SciPy, pandas, and matplotlib, and makes use of IPython and Jupyter (formerly called IPython Notebook) for the interface. NativeFit calculates simulated mass spectra based on input from the user and optimizes the parameters of the simulated spectrum to match experimental data. This serves both to visually deconvolute the data and to obtain the best estimates of mass, resolving power, average charge state, charge state width, and relative amounts of each analyte present.

4.3 Results and Discussion

4.3.1 Assigning Native Mass Spectra Using Simulated Spectra

In many of the samples, mixed stoichiometries and unexpected masses were observed, leading to congested spectra. Software was written to fit and annotate the data by simulating mass spectra based on user-input parameters, and then fitting the simulated spectra to the data by using a least-squares minimization. While starting parameters are provided manually,

each optimization is handled by the software. The software is quite flexible and remains very simple to use, especially for homomeric systems.

Several software tools have been developed to aid in assigning peaks to complex mass spectra. MassLynx (Waters Co.) includes peak assignment based on MaxEnt [173, 174], however the software does not calculate a simulated spectra so the fit cannot be visually compared with the data. Perhaps the most similar data fitting approach is that implemented by *Mass*ign [175]. *Mass*ign is a much more feature-rich tool that can automatically detect peaks and predict starting masses based on provided subunit masses. Other notable algorithms include Automass [176] and CHAMP [140]. Automass uses a game theory approach to fit complex native spectra with minimal user input. CHAMP is designed for fitting polydisperse systems such as small heat shock proteins [177,178], where a distribution of oligomeric states must be computed even before generating simulated component spectra. It should be noted however that given the wide charge state distributions that are closely spaced in m/z, even spectra such as ours require a fit method that can visually deconvolute, optimize simulated component spectra, and quantify relative quantities from the optimized component spectra. Further, none of the currently available tools are extensible or flexible, and are difficult to modify. The current ecosystem of mass spectral assignment tools leaves to be desired an option that is adaptable, interactive, and open source. We have developed NativeFit, a modular set of functions written in Python that provide a simple, command-based interface for fitting mass spectra.

The fit software was written in Python to take advantage of the many scientific libraries available and the high level nature of the language [145,179]. In addition to standard Python modules, our software relied on the free and open source modules SciPy, NumPy, pandas, and matplotlib [146, 180], and is able to run on any platform for which Python is available. The IPython library and Jupyter notebook provide an interface that is interactive and runs in a web browser, making it truly cross-platform. The advantage of the Jupyter notebook is that as the analysis is performed, each step is recorded along with its output to produce a notebook page. The notebook file can be saved for later use, retaining all of the interactivity

of the code that is included, as well as exported as either a PDF or static HTML file.

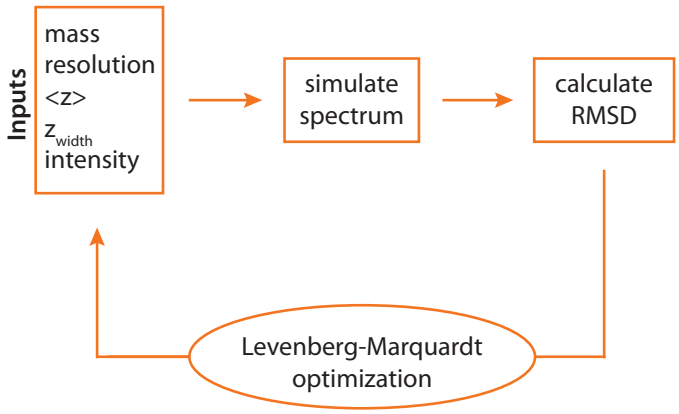


Figure 4.3: Initial guesses to the mass, peak resolution, average charge state, width of the charge-state distribution, and intensity of the charge-state distribution, are entered into the software as starting parameters. A simulated spectrum is calculated for each assembly, and the root-mean-square deviation (RMSD) is calculated from the difference between the sum of all simulated spectra and the experimental data. Using the Levenberg-Marquardt optimization as implemented in SciPy, the starting parameters are optimized so as to minimize the RMSD.

Briefly, an initial guess is made to the mass of the analyte m, the average charge state z_{avg}, the width of the charge state distribution z_{width}, the intensity of the distribution A, and the resolution of each peak m_{res} (Figure 4.3). It is assumed that the charge state distribution is Gaussian and that individual peaks are also Gaussian in shape. Initial guesses for the mass were based on the sequence mass and the hypothesis that hexamers would form, with later analyses also considering the possibility of pentamer forming. Theoretical m/z values that were calculated for various charge states of the expected hexamer and pentamer were matched up against peaks observed in the spectrum. Average charge states of each species were manually approximated based on observed relative peak intensities. Satisfactory initial guesses for these major species are often sufficient for optimizing the width and intensity of the distribution. To reduce computational expense and prevent the optimization from becoming stuck in local minima, the software only operates on one or two parameters at a

time until a reasonable fit is obtained, at which point the software can then optimize all the parameters to fit the data.

A typical workflow for the software begins with plotting the spectrum in an external window and visually identifying probable charge state distributions. The x- and y-values of the cursor position are displayed in the toolbar, and a list is constructed of the approximate m/z values belonging to the hypothesized charge state distribution. The function `peak_fit()` is run on these values and a table is returned with the best fit mass and approximate average charge state. These values can then automatically added to the list of potential charge state distributions using the `add_guess()` function. Alternatively, if a sequence mass is already known and average charge state estimated visually, m and z_{avg} can be added manually using the `add_assembly()` function. Default values are provided for m_{res}, z_{width}, and A, but these can also be manually specified. This is repeated for each charge state distribution that is immediately obvious to the researcher. The fit is then optimized by adjusting the parameters of the simulated spectrum to minimize the difference between simulated and experimental data. Typically the values for m and z_{avg} are held constant while the values for z_{width} and A are optimized, as the default values are less likely to match the true values. After optimizing, the optimized values are returned to the user and the new spectrum is plotted in the notebook. The researcher should at this point visually inspect the fits to determine if they have at least partially converged, and if so the optimization can be repeated holding z_{width} and A constant, allowing the values for mass and average charge state to better converge to the true value. Finally, the optimization can be repeated allowing allowing all the values to optimize with less chance of becoming stuck in local minima.

If only a fraction of assemblies were assigned during the first round of analysis, the visual output of simulated spectra may make additional distributions more obvious to the researcher, and the process of selecting peaks to obtain starting guesses for mass and charge state is repeated. Each step is recorded in the notebook as additional cells of either code or notes are added, and the workflow and logic of the researcher is conveniently recorded in the document that can either be distributed as a static report or an interactive, executable

notebook page for others to replicate or derive their own analyses from. Perhaps most importantly, the notebook format of our implementation allows the workflow itself and the thought process behind the analysis to be reproducible.

Use of the fit software allowed congested samples to be visually deconvoluted by first identifying and annotating expected species and determining accurate masses from the fit. Visual analysis of remaining peaks revealed additional potential species. Typically, if three or more peaks appeared at reasonable intensity and appeared to be part of a single Gaussian distribution, it was possible to approximate values for the starting masses and average charge states by visual inspection, and this distribution could be fit in a similar manner as the expected species.

Optimized fits typically reflected the experimentally observed spectrum well (Figure 4.4a–b). The difference between the experimental data and the sum of the simulated spectra acts as a satisfactory measure of the error of the fit.

For characterizing fusions proteins where multiple oligomeric states were observed, a key feature of this fit software was the ability to use the intensities of the optimized fits to directly determine the relative ratio of oligomers, the insight of which is described below in our results. Further, visualization of the expected fits quickly revealed errors in charge state assignments, preventing inaccurate mass determination.

4.3.2 Native Mass Spectrometry Results

Fusion proteins of $^{\Delta N1}GspE^{EpsE}$ and Hcp1 were connected using linkers that ranged from 3 to 8 amino acids (3aa–8aa) in length (Figure 4.4). All of these proteins assembled into higher order structures, forming a mixture of pentamers and hexamers. The 3aa and 4aa formed mostly pentamer (67% and 56% pentamer, respectively) and the 5aa formed approximately equal amounts of each (48% pentamer). Fusion proteins that had linkers of 6–8 amino acids in length formed hexamer exclusively. Relative quantities are based on the integration of fits to the native data as described in the experimental methods, and equal ionization efficiency of the pentamer and hexamer is assumed. Therefore it is likely that linker length

is the dominant factor in determining stoichiometry for these candidates, with longer linker lengths favoring the formation of hexamer. Shorter linker lengths provide less flexibility, and such fusion proteins may be too sterically hindered.

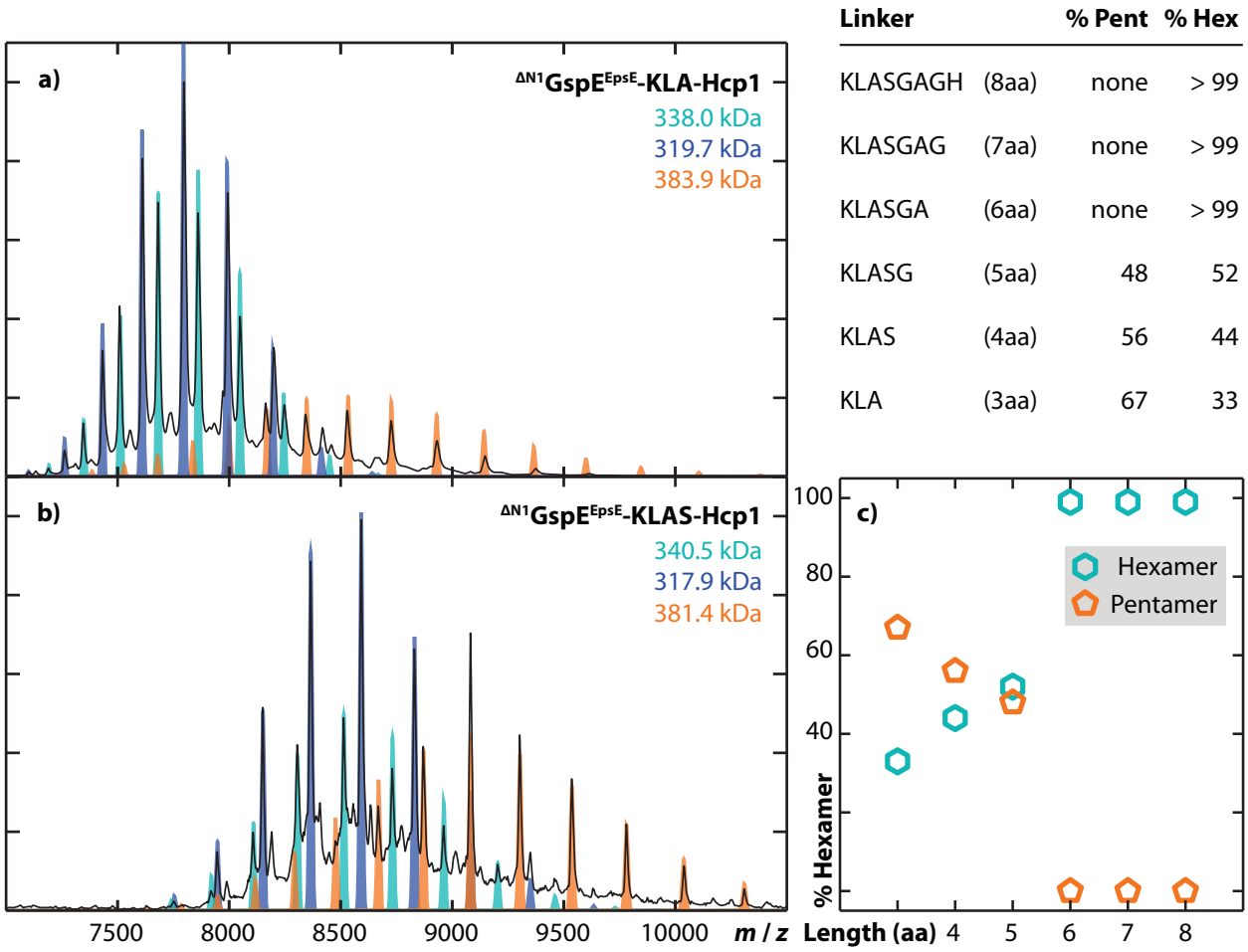


Figure 4.4: (**a**) The fitted native mass spectrum for $^{\Delta N1}$GspE^{EpsE}–KLA–Hcp1 shows good agreement between the experimentally determined masses and the expected masses (from the amino acid sequences) for both pentamer (m_{seq} = 318.9 kDa, m_{obs} = 319.7 kDa) and hexamer (m_{seq} = 382.7 kDa, m_{obs} = 383.9 kDa). (**b**) Fitted native mass spectra for $^{\Delta N1}$GspE^{EpsE}–KLAS–Hcp1. Fitted spectra agree well with the expected sequence masses for both pentamer (m_{seq} = 317.4 kDa, m_{obs} = 317.9 kDa) and hexamer (m_{seq} = 380.9 kDa, m_{obs} = 381.4 kDa). Increasing linker length favors formation of the hexamer, likely the effect of relieving steric crowding (**c**). Native MS of the other $^{\Delta N1}$GspE^{EpsE}–Hcp1 fusion protein complexes have been published previously [167] and are shown in Figure 4.8.

Certain spectra were not amenable to simple assignment using the sequence masses. In Figure 4.5a the native MS of $^{\Delta N1}$GspE^{EpsE}–KLASGA–Hcp1 reveals two components. The blue fit corresponds to the hexamer (m_{seq} = 384.0 kDa, m_{obs} = 384.6 kDa), however an additional mass is seen at higher m/z, with an approximate fit in orange. This distribution (orange) has approximately half the number of charge states as the hexamer. Confidence in charge state assignment increases with the number of charge states, however the closer the charge states are in m/z space, which is often the case for massive ions of high charge, the more ambiguous the charge state assignment can become. In Figure 4.5b–c, the same native mass spectrum of the unidentified complex is shown with two different charge state assignments, giving an m_{obs} of 436.3 kDa for the spectrum in Figure 4.5b and an mobs of 447.0 kDa for the spectrum in Figure 4.5c. Using the presumed charge state assignments and masses, simulated spectra (orange) were optimized to fit each spectrum. The fits are indistinguishable, and therefore it is impossible to say with confidence whether the mass difference between the hexamer and additional conformer is 51.5 kDa or 61.4 kDa. Additional charge states are necessary for proper assignment, however the complex was resistant to CID. Even at high activation conditions, $^{\Delta N1}$GspE^{EpsE}–KLASGA–Hcp1 pentamer was the only observed CID product.

4.3.3 Charge Reduction

In order to unambiguously assign charge states to the native mass spectra in Figure 4.5b–c, we have implemented CAPTR to reduce the charge of the native-like protein complexes via ion/ion proton transfer reactions [171]. In these experiments, native-like protein complex cations are transmitted through a cloud of trapped singly-charged anions to reduce the charge on the protein complexes. The resulting charge-reduced CAPTR products may then be used to assign charge to the precursor ions.

For CAPTR, cations above ~8 000 m/z were transmitted into the trap cell, which includes protein charge states at 10 670 m/z and 10 950 m/z as well as any salt adducts in that m/z range. The corresponding CAPTR spectrum is shown in Figure 4.6, which

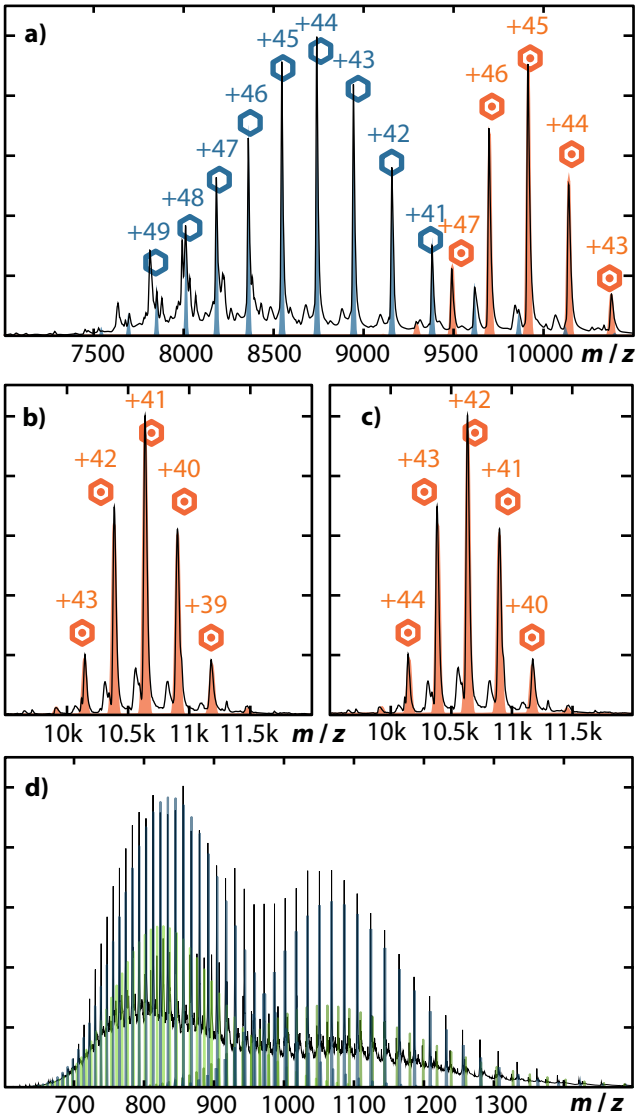


Figure 4.5: Native mass spectrum of $^{\Delta N1}$GspE^{EpsE}–KLASGA–Hcp1 (**a**). The fit spectrum based on the hexamer sequence mass (m_{seq} = 384.0 kDa, m_{obs} = 384.6 kDa) is shown in blue. Shown in orange is the fit spectrum to the hexamer plus an unidentified mass (Δm_{obs} = 71.5 kDa). Two potential charge state assignments are show in (**b**) and (**c**) centered at either +41 and +42 with ambiguous fits. The sample was denatured and electrosprayed (**d**), revealing two mass species. The light blue fit spectrum corresponds to the denatured $^{\Delta N1}$GspE^{EpsE}–6aa–Hcp1 monomer (m_{seq} = 64.0 kDa, m_{obs} = 64.1 kDa), whereas the green fit spectrum is an unidentified protein of mass 74.3 kDa, which agrees well the mass of bifunctional polymyxin resistance protein ArnA (m_{seq} = 74.3 kDa).

shows unreacted precursor ions, and charge-reduced CAPTR product peaks from 25 000 to 55 000 m/z. Interestingly, salt clusters, which commonly lower the signal-to-noise of native MS experiments, have been separated from protein complex ions via charge reduction and are observed from 12 000 to 24 000 m/z. The m/z center of the CAPTR product peaks were then multiplied by candidate charge state assignments, and the average of the corresponding mass values were plotted against their standard deviations in Figure 4.7. Note that when this approach was used with only the native charge states, the mass and charge assignments were ambiguous, but when the lower CAPTR product charge states are considered the mass may be unambiguously assigned as 449.2 kDa.

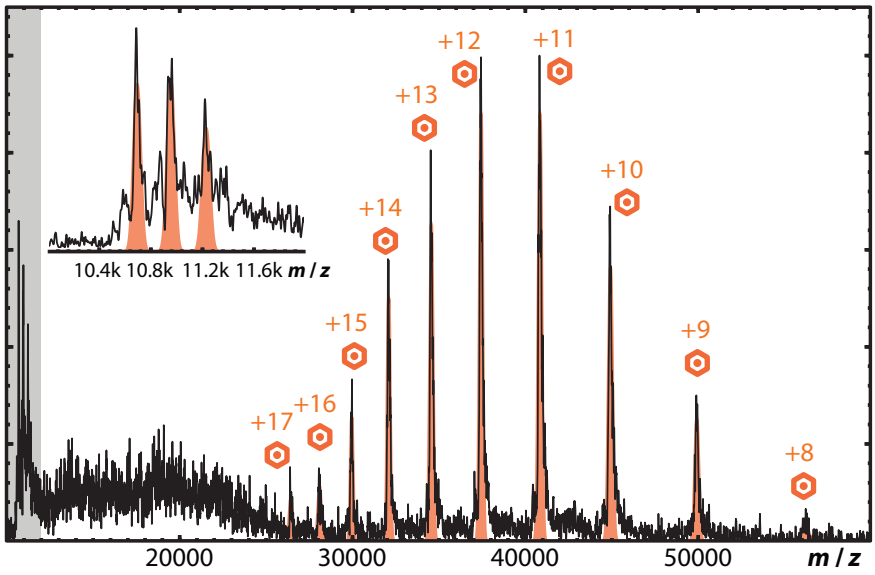


Figure 4.6: CAPTR spectrum of $^{\Delta N1}$GspE^{EpsE}–KLASGA–Hcp1. Shown in the upper left corner is an enlargement of the peaks corresponding to the precursor (highlighted in grey). The observed charge state distribution reflects a loss of approximately 58–81% of the initial positive charge. The increase in number of charge states, along with the increased spacing at high m/z, allow unambiguous charge state assignment and mass determination (m_{obs} = 449.2 kDa).

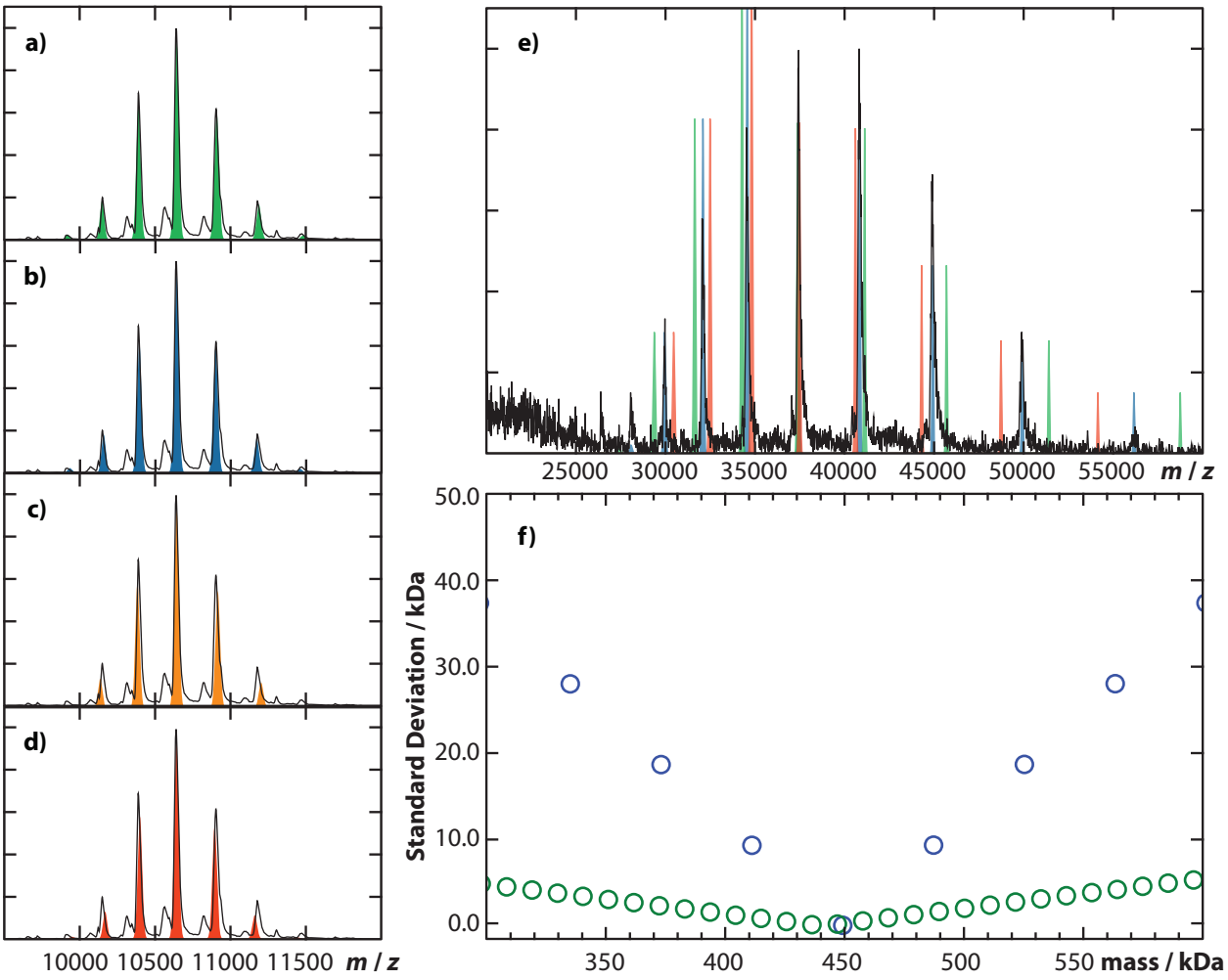


Figure 4.7: The native mass spectrum of $^{\Delta N1}$GspE^{EpsE}–KLASGA–Hcp1 is shown with four different simulated fits. Each fit assumes a different charge state assignment, each differing by a single charge, and the corresponding mass (**a**) 436.3 kDa, (**b**) 447.0 kDa, (**c**) 425.7 kDa, and (**d**) 457.6 kDa. Visually, the four fit spectra are very similar and determining the correct one is challenging if not impossible. Charge state assignment of the CAPTR spectrum is much less ambiguous (**e**). The blue fit shows the correct charge state assignment. Shifting the assignment by only a single charge state (red, 487.3 kDa, +1; green, 411.2 kDa, -1) is clearly penalizing. The ambiguity in charge state assignment, however, can be quantified (**f**). For any given charge state assignment, the detected centroids can be multiplied by their charge states to obtain the corresponding mass ($M = m * z - zH$), and the average should be reported ($\langle M \rangle$). For the correct charge state assignment, the values of M should be approximately the same, i.e. the standard deviation (σ_M) should be low. Plotted in (**f**) are the average masses $\langle M \rangle$ determined from a given charge assignment and the corresponding σ_M (lower is better). For systems with greater numbers of peaks at high m/z, such as CAPTR, an incorrect charge state assignment is much more heavily penalized in terms of standard deviation. Additionally, the change in mass is much greater for a single charge state shift.

4.3.4 Analysis of Protein Monomers

In the case of protein complexes that were assigned masses that did not agree well with those based on the expected sequences, C4 Zip Tips (Millipore) were used to denature the protein and acquire accurate masses of the individual monomers. C4 Zip Tips are micropipette tips containing a C4 resin to which the proteins adhere, immobilizing them so that salts can be washed away. Protocols for zip tips are included with the product, however a few modifications were made to ensure compatibility with our system. The protein was initially denatured and acidified with aqueous 1M GdCl / 1.0% TFA. It is important to note that it is necessary that the protein not yet been exchanged into ammonium acetate, as the acetate acts as a buffer and will prevent the sample from being fully acidified. After being loaded onto the resin, the sample is then washed with aqueous 0.1% TFA and eluted with 75/24/1 acetonitrile/water/formic acid and is electrosprayed from this solution. The sample was analyzed with gentle activating conditions and the spectrum in Figure 4.5d was obtained. Two distributions were observed, a species corresponding to the $^{\Delta N1}GspE^{EpsE}$ monomer (m_{seq} = 64.00 kDa, m_{obs} = 64.13 kDa) and an additional protein of mass 74.33 kDa. This matches the mass of bifunctional polymyxin resistance protein ArnA to 614 ppm, which is known to be found in the strain of *E. coli* used for expression and was additionally detected using bottom-up proteomics (5 peptides, 8.03% sequence coverage).

The results of CAPTR have provided an unambiguous charge state assignment for the distribution described in Figure 4.5a–c, allowing a confident mass determination of 446.18 kDa. Using the monomer mass determined from the Zip Tip to calculate the hexamer mass gives a mass of 384.764 kDa (m_{seq} = 384.025 kDa). The difference in m_{obs} is therefore 62.45 kDa. The identity of this mass shift was not further investigated, however it is interesting to note that the likelihood of a random protein copurifying and forming a complex with the fusion complex is low. We would therefore propose that the available evidence suggests the mass increase is due to binding by a molecular chaperone. Firstly, fusion proteins are engineered proteins, and as such it is a very real possibility that their structure may be less ordered

than the wild-type counterparts of their constituents, making them an attractive target for chaperones, which bind such proteins to prevent precipitation. Chaperones are also promiscuous binders — they will effectively bind to many proteins that are sufficiently disordered. Further, the stabilizing effects of chaperone-binding would likely interfere with CID, as some degree of unfolding during gas-phase activation must occur in order for subunits to dissociate. One possible identity of this protein could be Heat Shock Protein 70 (Uniprot: C6EK46, m_{seq} = 61 946 Da) with a mass error of 3 366 ppm.

4.4 Conclusions

We have presented an application of native mass spectrometry as a rapid characterization method for engineered protein complexes. Protein complexes such as fusion proteins, which will not have a wild-type homologue, do not have necessarily predictable folding and assembly behavior. For proteins that can successfully fold and assemble into complexes, traditional methods such as SEC are not sufficiently resolving to separate close oligomeric states whereas end-goal techniques such as crystallization are too time consuming. Native mass-spectrometry can provide same-day results on the oligomeric states of formed complexes, allowing potential candidates for crystallization to be quickly identified.

Our method of spectral annotation, using in-house software, is both simple to use and flexible. Initial guesses about complex masses can be determined from sequence and the theoretical m/z values calculated before fitting the data. For well-resolved spectra, identification of m/z shifts are straightforward, and peaks can be matched up to the corresponding complex revealing approximate mass shifts and allowing better initial estimates for mass. Intensities of these peaks can be visually compared and an approximate average charge-state determined. With good starting guesses and by restricting parameters, fits can be obtained in minutes. The annotation that results from use of this software and the output of the least-squares optimization gives the researcher both a visual qualitative score and a quantitative score evaluating the fit, allowing unexpected masses and determine relative quantities of oligomeric states to be identified. This method of rapidly characterizing fusion

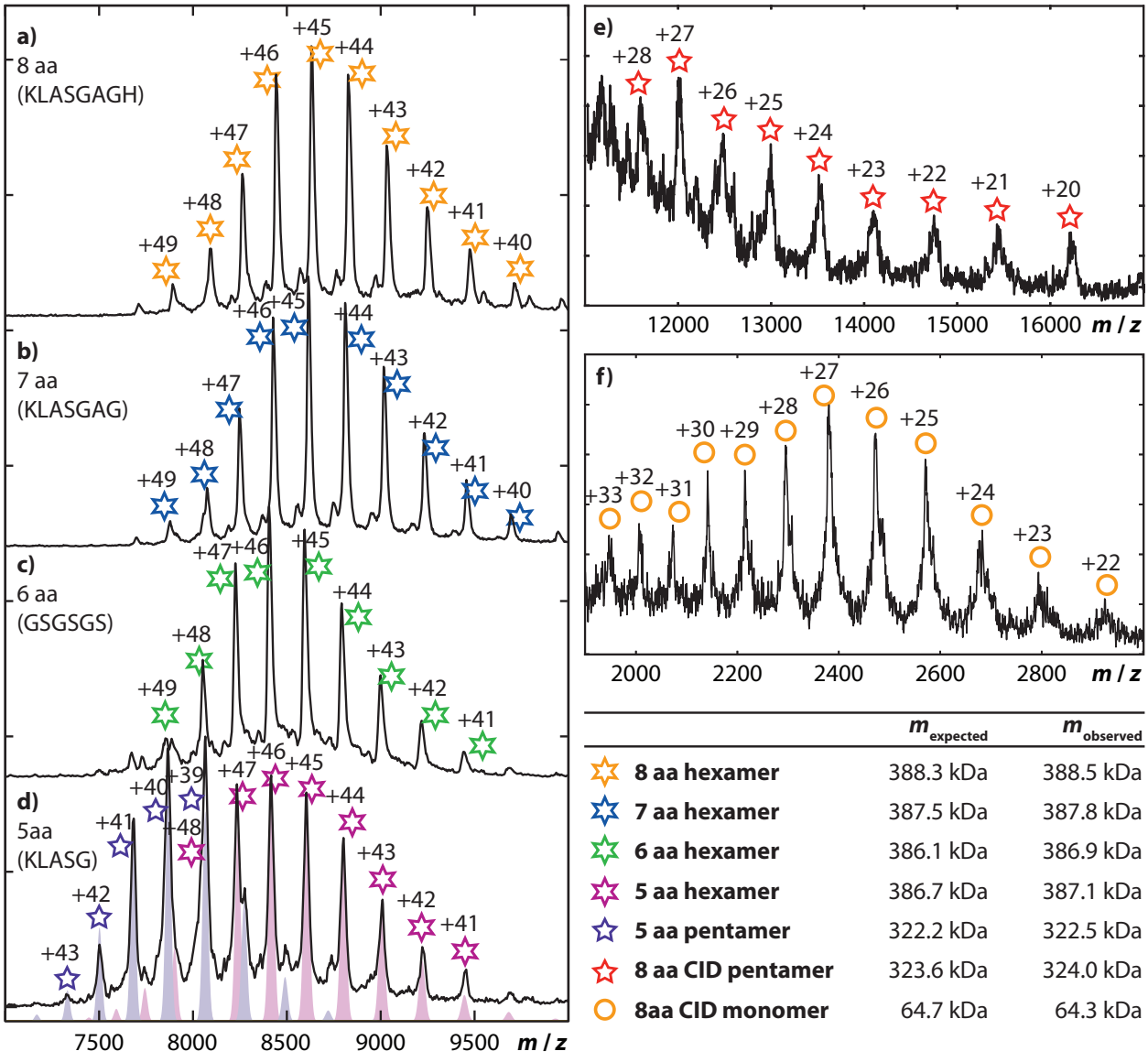


Figure 4.8: The data show that $^{\Delta N1}$GspE^{EpsE}–8aa–Hcp1 (**a**), $^{\Delta N1}$GspE^{EpsE}–7aa–Hcp1 (**b**), and $^{\Delta N1}$GspE^{EpsE}–6aa–Hcp1 (**c**) can each assemble as hexamers in solution, but that $^{\Delta N1}$GspE^{EpsE}–5aa–Hcp1 (**d**) forms both pentameric and hexameric assemblies. Tandem mass spectra were also performed in which all ions greater than ~7500 m/z were isolated in the gas phase and subsequently fragmented using collision-induced dissociation. The appearance of pentamer (**e**) and monomer (**f**) product ions during all gas-phase CID experiments supports the assignment of hexamer precursor ions in **a**–**d**. Note that the loss of peptide ions from the precursor ions was also observed, but that fragmentation channel was less structurally informative. The measured and expected masses for all ions are reported in the table. This figure is reproduced from Lu et al. [167].

proteins saves time and resources by determining which fusion proteins are viable candidates for crystallization, and provides an unmatched level of detail compared to solution-phase techniques.

With complementary mass spectrometry techniques available to answer additional questions raised by unexpected masses, such as contaminant proteins or missing residues, there is sufficient feedback to optimize fusion protein sequence and expression methods. This method has been proven through the crystallization of fusion protein complexes [167] — the published results are reproduced here for reference in Figure 4.8 — and provides a framework for tackling similarly challenging systems as engineered proteins become increasingly a target of interest for structural biology.

4.5 Acknowledgements

S.T.M. thanks Dr. Martin Sadilek (University of Washington) for support, training, and assistance with LC-MS experiments. S.T.M. and M.F.B. acknowledge support from the University of Washington. Research reported in this publication was supported by NIAID of the National Institutes of Health under award number AI34501 to W.G.J.H.

BIBLIOGRAPHY

[1] John B. Fenn, Matthias Mann, Chin Kai Meng, Shek Fu Wong, and Craig M. Whitehouse. Electrospray ionization for mass spectrometry of large biomolecules. *Science*, 246(4926):64–71, **1989**.

[2] John B. Fenn. Electrospray wings for molecular elephants (nobel lecture). *Angewandte Chemie International Edition*, 42(33):3871–3894, **2003**.

[3] Matthias Wilm. Principles of electrospray ionization. *Molecular & Cellular Proteomics*, 10(7):M111.009407, **2011**.

[4] Ruedi Aebersold and Matthias Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928):198–207, **2003**.

[5] Matthias Mann and Ole N. Jensen. Proteomic analysis of post-translational modifications. *Nature Biotechnology*, 21(3):255–261, **2003**.

[6] Richard D. Smith, Joseph A. Loo, Charles G. Edmonds, Charles J. Barinaga, and Harold R. Udseth. New developments in biochemical mass spectrometry: electrospray ionization. *Analytical Chemistry*, 62(9):882–899, **1990**.

[7] Brandon T. Ruotolo, Kevin Giles, Iain Campuzano, Alan M. Sandercock, Robert H. Bateman, and Carol V. Robinson. Evidence for macromolecular protein rings in the absence of bulk water. *Science*, 310(5754):1658–1661, **2005**.

[8] Robert H.H. van den Heuvel and Albert J.R. Heck. Native protein mass spectrometry: from intact oligomers to functional machineries. *Current Opinion in Chemical Biology*, 8(5):519–526, **2004**.

[9] Justin L.P. Benesch and Brandon T. Ruotolo. Mass spectrometry: come of age for structural and dynamical biology. *Current Opinion in Structural Biology*, 21(5):641–649, **2011**.

[10] Gillian R. Hilton and Justin L. P. Benesch. Two decades of studying non-covalent biomolecular assemblies by means of electrospray ionization mass spectrometry. *Journal of the Royal Society Interface*, 9(70):801–816, **2012**.

[11] Albert J. R. Heck and Robert H. H. van den Heuvel. Investigation of intact protein complexes by mass spectrometry. *Mass Spectrometry Reviews*, 23(5):368–389, **2004**.

[12] Justin L.P. Benesch, Brandon T. Ruotolo, Douglas A. Simmons, and Carol V. Robinson. Protein complexes in the gas phase: technology for structural genomics and proteomics. *Chemical Reviews*, 107(8):3544–3567, **2007**.

[13] Joseph A. Loo. Studying noncovalent protein complexes by electrospray ionization mass spectrometry. *Mass Spectrometry Reviews*, 16(1):1–23, **1997**.

[14] Joseph A. Loo. Electrospray ionization mass spectrometry: a technology for studying noncovalent macromolecular complexes. *International Journal of Mass Spectrometry*, 200(1–3):175–186, **2000**.

[15] Michal Sharon. Structural MS pulls its weight. *Science*, 340(6136):1059–1060, **2013**.

[16] Lars Konermann and D.J. Douglas. Unfolding of proteins monitored by electrospray ionization mass spectrometry: a comparison of positive and negative ion modes. *Journal of the American Society for Mass Spectrometry*, 9(12):1248–1254, **1998**.

[17] Joost Snijder, Rebecca J. Rose, David Veesler, John E. Johnson, and Albert J. R. Heck. Studying 18 MDa virus assemblies with native mass spectrometry. *Angewandte Chemie International Edition*, 52(14):4020–4023, **2013**.

[18] Helena Hernández and Carol V. Robinson. Determining the stoichiometry and interactions of macromolecular assemblies from mass spectrometry. *Nature Protocols*, 2(3):715–726, **2007**.

[19] Albert J.R. Heck. Native mass spectrometry: a bridge between interactomics and structural biology. *Nature Methods*, 5(11):927–933, **2008**.

[20] Justin L.P. Benesch, J. Andrew Aquilina, Brandon T. Ruotolo, Frank Sobott, and Carol V. Robinson. Tandem mass spectrometry reveals the quaternary organization of macromolecular assemblies. *Chemistry & Biology*, 13(6):597–605, **2006**.

[21] Eugen Damoc, Christopher S. Fraser, Min Zhou, Hortense Videler, Greg L. Mayeur, John W. B. Hershey, Jennifer A. Doudna, Carol V. Robinson, and Julie A. Leary. Structural characterization of the human eukaryotic initiation factor 3 protein complex by mass spectrometry. *Molecular & Cellular Proteomics*, 6(7):1135–1146, **2007**.

[22] Justin L.P. Benesch. Collisional activation of protein complexes: Picking up the pieces. *Journal of the American Society for Mass Spectrometry*, 20(3):341–348, **2009**.

[23] Michael W. Senko, J. Paul Speir, and Fred W. McLafferty. Collisional activation of large multiply charged ions using fourier transform mass spectrometry. *Analytical Chemistry*, 66(18):2801–2808, **1994**.

[24] Frank Sobott and Carol V. Robinson. Characterising electrosprayed biomolecules using tandem-MS—the noncovalent GroEL chaperonin assembly. *International Journal of Mass Spectrometry*, 236(1–3):25–32, **2004**.

[25] Geoffrey Taylor. Disintegration of water drops in an electric field. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 280(1382):383–397, **1964**.

[26] Matthias S. Wilm and Matthias Mann. Electrospray and Taylor-cone theory, Dole's beam of macromolecules at last? *International Journal of Mass Spectrometry and Ion Processes*, 136(2):167–180, **1994**.

[27] J. Fernandez de la Mora. Electrospray ionization of large multiply charged species proceeds via Dole's charged residue mechanism. *Analytica Chimica Acta*, 406(1):93–104, **2000**.

[28] Lord Rayleigh. On the equilibrium of liquid conducting masses charged with electricity. *Philosophical Magazine*, 14(87):184–186, **1882**.

[29] J.M.H. Peters. Rayleigh's electrified water drops. *European Journal of Physics*, 1(3):143, **1980**.

[30] Malcolm Dole, L. L. Mack, R. L. Hines, R. C. Mobley, L. D. Ferguson, and M. B. Alice. Molecular beams of macroions. *The Journal of Chemical Physics*, 49(5):2240–2249, **1968**.

[31] Alessandro Gomez and Keqi Tang. Charge and fission of droplets in electrostatic sprays. *Physics of Fluids*, 6(1):404–414, **1994**.

[32] Matthias Wilm and Matthias Mann. Analytical properties of the nanoelectrospray ion source. *Analytical Chemistry*, 68(1):1–8, **1996**.

[33] Kimberly L. Davidson, Derek R. Oberreit, Christopher J. Hogan Jr., and Matthew F. Bush. Droplet sizes, ionization currents, and nonspecific aggregation in native electrokinetic electrospray ionization. *Analytical Chemistry*, **Submitted**.

[34] Andrea Schmidt, Michael Karas, and Thomas Dülcks. Effect of different solution flow rates on analyte ion signals in nano-ESI MS, or: when does ESI turn into nano-ESI? *Journal of the American Society for Mass Spectrometry*, 14(5):492–500, **2003**.

[35] Arno Wortmann, Anna Kistler-Momotova, Renato Zenobi, Martin C. Heine, Oliver Wilhelm, and Sotiris E. Pratsinis. Shrinking droplets in electrospray ionization and their influence on chemical equilibria. *Journal of the American Society for Mass Spectrometry*, 18(3):385–393, **2007**.

[36] M. Scott Kriger, Kelsey D. Cook, and Roswitha S. Ramsey. Durable gold-coated fused silica capillaries for use in electrospray mass spectrometry. *Analytical Chemistry*, 67(2):385–389, **1995**.

[37] Graham T.T. Gibson, Samuel M. Mugo, and Richard D. Oleschuk. Nanoelectrospray emitters: Trends and perspective. *Mass Spectrometry Reviews*, 28(6):918–936, **2009**.

[38] Anatoli N. Verentchikov, Werner Ens, and Kenneth G. Standing. Reflecting time-of-flight mass spectrometer with an electrospray ion source and orthogonal extraction. *Analytical Chemistry*, 66(1):126–133, **1994**.

[39] Frank Sobott, Helena Hernández, Margaret G. McCammon, Mark A. Tito, and Carol V. Robinson. A tandem mass spectrometer for improved transmission and analysis of large macromolecular assemblies. *Analytical Chemistry*, 74(6):1402–1407, **2002**.

[40] Robert H.H. van den Heuvel, Esther van Duijn, Hortense Mazon, Silvia A. Synowsky, Kristina Lorenzen, Cees Versluis, Stan J.J. Brouns, Dave Langridge, John van der Oost, John Hoyes, and Albert J.R. Heck. Improving the performance of a quadrupole time-of-flight instrument for macromolecular mass spectrometry. *Analytical Chemistry*, 78(21):7473–7483, **2006**.

[41] V.I. Kozlovski, L.J. Donald, V.M. Collado, V. Spicer, A.V. Loboda, I.V. Chernushevich, W. Ens, and K.G. Standing. A TOF mass spectrometer for the study of noncovalent complexes. *International Journal of Mass Spectrometry*, 308(1):118–125, **2011**.

[42] Cherokee S. Hoaglund, Stephen J. Valentine, C. Ray Sporleder, James P. Reilly, , and David E. Clemmer. Three-dimensional ion mobility/TOFMS analysis of electrosprayed biomolecules. *Analytical Chemistry*, 70(11):2236–2242, **1998**.

[43] Steven D. Pringle, Kevin Giles, Jason L. Wildgoose, Jonathan P. Williams, Susan E. Slade, Konstantinos Thalassinos, Robert H. Bateman, Michael T. Bowers, and James H. Scrivens. An investigation of the mobility separation of some peptide and protein ions using a new hybrid quadrupole/travelling wave IMS/oa-ToF instrument. *International Journal of Mass Spectrometry*, 261(1):1–12, **2007**.

[44] Christine Eckers, Alice M.-F. Laures, Kevin Giles, Hilary Major, and Steve Pringle. Evaluating the utility of ion mobility separation in combination with high-pressure liquid chromatography/mass spectrometry to facilitate detection of trace impurities in formulated drug products. *Rapid Communications in Mass Spectrometry*, 21(7):1255–1263, **2007**.

[45] Kevin Giles, Steven D. Pringle, Kenneth R. Worthington, David Little, Jason L. Wildgoose, and Robert H. Bateman. Applications of a travelling wave-based radio-frequency-only stacked ring ion guide. *Rapid Communications in Mass Spectrometry*, 18(20):2401–2414, **2004**.

[46] Brandon T. Ruotolo, Justin L.P. Benesch, Alan M. Sandercock, Suk-Joon Hyung, and Carol V. Robinson. Ion mobility-mass spectrometry analysis of large protein complexes. *Nature Protocols*, 3(7):1139–1152, **2008**.

[47] Matthew F. Bush, Zoe Hall, Kevin Giles, John Hoyes, Carol V. Robinson, and Brandon T. Ruotolo. Collision cross sections of proteins and their complexes: A calibration framework and database for gas-phase structural biology. *Analytical Chemistry*, 82(22):9557–9565, **2010**.

[48] Kevin Giles, Jonathan P. Williams, and Iain Campuzano. Enhancements in travelling wave ion mobility resolution. *Rapid Communications in Mass Spectrometry*, 25(11):1559–1566, **2011**.

[49] Samuel J. Allen, Samuel T. Marionni, K. Giles, T. Gilbert, and Matthew F. Bush. Design and characterization of a new ion mobility cell for protein complexes. In *60th American Society for Mass Spectrometry Conference*. Vancouver, BC, **2012**.

[50] Richard D. Smith, Joseph A. Loo, Charles J. Barinaga, Charles G. Edmonds, and Harold R. Udseth. Collisional activation and collision-activated dissociation of large multiply charged polypeptides and proteins produced by electrospray ionization. *Journal of the American Society for Mass Spectrometry*, 1(1):53–65, **1990**.

[51] P.B. Armentrout, Kent M. Ervin, and M.T. Rodgers. Statistical rate theory and kinetic energy-resolved ion chemistry: Theory and applications. *The Journal of Physical Chemistry A*, 112(41):10071–10085, **2008**.

[52] John C. Jurchen and Evan R. Williams. Origin of asymmetric charge partitioning in the dissociation of gas-phase protein homodimers. *Journal of the American Chemical Society*, 125(9):2817–2826, **2003**.

[53] Stephen V. Sciuto, Jiangjiang Liu, and Lars Konermann. An electrostatic charge partitioning model for the dissociation of protein complexes in the gas phase. *Journal of the American Society for Mass Spectrometry*, 22(10):1679–1689, **2011**.

[54] Michal Sharon, Thomas Taverner, Xavier I. Ambroggio, Raymond J. Deshaies, and Carol V. Robinson. Structural organization of the 19S proteasome lid: Insights from MS of intact complexes. *PLoS Biology*, 4(8):e267, **2006**.

[55] Abu B. Kanu, Prabha Dwivedi, Maggie Tam, Laura Matz, and Herbert H. Hill. Ion mobility—mass spectrometry. *Journal of Mass Spectrometry*, 43(1):1–22, **2008**.

[56] David E. Clemmer and Martin F. Jarrold. Ion mobility measurements and their applications to clusters and biomolecules. *Journal of Mass Spectrometry*, 32(6):577–592, **1997**.

[57] M. F. Mesleh, J. M. Hunter, A. A. Shvartsburg, G. C. Schatz, and M. F. Jarrold. Structural information from ion mobility measurements: effects of the long-range potential. *The Journal of Physical Chemistry*, 100(40):16082–16086, **1996**.

[58] Thomas Wyttenbach, Christian Bleiholder, and Michael T. Bowers. Factors contributing to the collision cross section of polyatomic ions in the kilodalton to gigadalton range: Application to ion mobility measurements. *Analytical Chemistry*, 85(4):2191–2199, **2013**.

[59] Ewa Jurneczko and Perdita E. Barran. How useful is ion mobility mass spectrometry for structural biology? The relationship between protein crystal structures and their collision cross sections in the gas phase. *Analyst*, 136:20–28, **2011**.

[60] Edward A. Mason and Earl W. McDaniel. *Transport Properties of Ions in Gases*. Wiley, New York, **1988**.

[61] Kathrin Breuker and Fred W. McLafferty. Stepwise evolution of protein native structure with electrospray into the gas phase, 10^{-12} to 10^{2} s. *Proceedings of the National Academy of Sciences*, 105(47):18145–18152, **2008**.

[62] Edward Mack. Average cross-sectional areas of molecules by gaseous diffusion methods. *Journal of the American Chemical Society*, 47(10):2468–2482, **1925**.

[63] Gert von Helden, Ming-Teh Hsu, Nigel Gotts, and Michael T. Bowers. Carbon cluster cations with up to 84 atoms: structures, formation mechanism, and reactivity. *The Journal of Physical Chemistry*, 97(31):8182–8192, **1993**.

[64] Alexandre A. Shvartsburg and Martin F. Jarrold. An exact hard-spheres scattering model for the mobilities of polyatomic ions. *Chemical Physics Letters*, 261(1–2):86–91, **1996**.

[65] Alexandre A. Shvartsburg, Stefan V. Mashkevich, Erin Shammel Baker, and Richard D. Smith. Optimization of algorithms for ion mobility calculations. *The Journal of Physical Chemistry A*, 111(10):2002–2010, **2007**.

[66] Alexandre A. Shvartsburg, George C. Schatz, and Martin F. Jarrold. Mobilities of carbon cluster ions: Critical importance of the molecular attractive potential. *The Journal of Chemical Physics*, 108(6):2416–2423, **1998**.

[67] Thomas Wyttenbach, Gert von Helden, Joseph J. Batka Jr., Douglas Carlat, and Michael T. Bowers. Effect of the long-range potential on ion mobility measurements. *Journal of the American Society for Mass Spectrometry*, 8(3):275–282, **1997**.

[68] Erik G. Marklund, Matteo T. Degiacomi, Carol V. Robinson, Andrew J. Baldwin, and Justin L.P. Benesch. Collision cross sections for structural proteomics. *Structure*, 23(4):791–799, **2015**.

[69] Christian Bleiholder, Thomas Wyttenbach, and Michael T. Bowers. A novel projection approximation algorithm for the fast and accurate computation of molecular collision cross sections (I). Method. *International Journal of Mass Spectrometry*, 308(1):1–10, **2011**.

[70] Christian Bleiholder, Stephanie Contreras, Thanh D. Do, and Michael T. Bowers. A novel projection approximation algorithm for the fast and accurate computation of molecular collision cross sections (II). Model parameterization and definition of empirical shape factors for proteins. *International Journal of Mass Spectrometry*, 345–347:89–96, **2013**.

[71] Stanley E. Anderson, Christian Bleiholder, Erin R. Brocker, Peter J. Stang, and Michael T. Bowers. A novel projection approximation algorithm for the fast and accurate computation of molecular collision cross sections (III): Application to supramolecular coordination-driven assemblies with complex shapes. *International Journal of Mass Spectrometry*, 330–332:78–84, **2012**.

[72] Matthew F. Bush, Zoe Hall, Argyris Politis, Daniel Barsky, and Carol V. Robinson. Interpreting the collision cross sections of protein complexes: Models, approximations, errors, and best practices. In *59th American Society for Mass Spectrometry Conference*. Denver, CO, **2011**.

[73] Nenad Ban, Poul Nissen, Jeffrey Hansen, Peter B. Moore, and Thomas A. Steitz. The complete atomic structure of the large ribosomal subunit at 2.4 Å resolution. *Science*, 289(5481):905–920, **2000**.

[74] Gongyi Zhang, Elizabeth A Campbell, Leonid Minakhin, Catherine Richter, Konstantin Severinov, and Seth A Darst. Crystal structure of *Thermus aquaticus* core RNA polymerase at 3.3 Å resolution. *Cell*, 98(6):811–824, **1999**.

[75] Maurizio Pellecchia, Daniel S. Sem, and Kurt Wuthrich. NMR in drug discovery. *Nature Reviews Drug Discovery*, 1(3):211–219, **2002**.

[76] Tamiji Nakanishi, Mayumi Miyazawa, Masayoshi Sakakura, Hiroaki Terasawa, Hideo Takahashi, and Ichio Shimada. Determination of the interface of a large protein complex by transferred cross-saturation measurements. *Journal of Molecular Biology*, 318(2):245–249, **2002**.

[77] Andrej Sali, Robert Glaeser, Thomas Earnest, and Wolfgang Baumeister. From words to literature in structural proteomics. *Nature*, 422(6928):216–225, **2003**.

[78] Frank Alber, Svetlana Dokudovskaya, Liesbeth M. Veenhoff, Wenzhu Zhang, Julia Kipper, Damien Devos, Adisetyantari Suprapto, Orit Karni-Schmidt, Rosemary Williams, Brian T. Chait, Michael P. Rout, and Andrej Sali. Determining the architectures of macromolecular assemblies. *Nature*, 450(7170):683–694, **2007**.

[79] Thomas Walzthoeni, Alexander Leitner, Florian Stengel, and Ruedi Aebersold. Mass spectrometry supported determination of protein complex structure. *Current Opinion in Structural Biology*, 23(2):252–260, **2013**.

[80] Suk-Joon Hyung and Brandon T. Ruotolo. Integrating mass spectrometry of intact protein complexes into structural proteomics. *PROTEOMICS*, 12(10):1547–1564, **2012**.

[81] Frank Sobott and Carol V. Robinson. Protein complexes gain momentum. *Current Opinion in Structural Biology*, 12(6):729–734, **2002**.

[82] Thomas Taverner, Helena Hernández, Michal Sharon, Brandon T. Ruotolo, Dijana Matak-Vinković, Damien Devos, Robert B. Russell, and Carol V. Robinson. Subunit architecture of intact protein complexes from mass spectrometry and homology modeling. *Accounts of Chemical Research*, 41(5):617–627, **2008**.

[83] Helena Hernández, Andrzej Dziembowski, Thomas Taverner, Bertrand Séraphin, and Carol V. Robinson. Subunit architecture of multimeric complexes isolated directly from cells. *EMBO reports*, 7(6):605–610, **2006**.

[84] Min Zhou, Alan M. Sandercock, Christopher S. Fraser, Gabriela Ridlova, Elaine Stephens, Matthew R. Schenauer, Theresa Yokoi-Fong, Daniel Barsky, Julie A. Leary, John W. Hershey, Jennifer A. Doudna, and Carol V. Robinson. Mass spectrometry reveals modularity and a complete subunit interaction map of the eukaryotic translation factor eIF3. *Proceedings of the National Academy of Sciences*, 105(47):18139–18144, **2008**.

[85] Yueyang Zhong, Jun Feng, and Brandon T. Ruotolo. Robotically assisted titration coupled to ion mobility-mass spectrometry reveals the interface structures and analysis parameters critical for multiprotein topology mapping. *Analytical Chemistry*, 85(23):11360–11368, **2013**.

[86] Tara L. Pukala, Brandon T. Ruotolo, Min Zhou, Argyris Politis, Raluca Stefanescu, Julie A. Leary, and Carol V. Robinson. Subunit architecture of multiprotein assemblies determined using restraints from gas-phase measurements. *Structure*, 17(9):1235–1243, **2009**.

[87] Argyris Politis, Florian Stengel, Zoe Hall, Helena Hernandez, Alexander Leitner, Thomas Walzthoeni, Carol V. Robinson, and Ruedi Aebersold. A mass spectrometry-based hybrid method for structural modeling of protein complexes. *Nature Methods*, 11(4):403–406, **2014**.

[88] Zoe Hall, Argyris Politis, and Carol V. Robinson. Structural modeling of heteromeric protein complexes from disassembly pathways and ion mobility-mass spectrometry. *Structure*, 20(9):1596–1609, **2012**.

[89] Argyris Politis, Ah Young Park, Zoe Hall, Brandon T. Ruotolo, and Carol V. Robinson. Integrative modelling coupled with ion mobility mass spectrometry reveals structural features of the clamp loader in complex with single-stranded DNA binding protein. *Journal of Molecular Biology*, 425(23):4790–4801, **2013**.

[90] Argyris Politis, Carla Schmidt, Elina Tjioe, Alan M. Sandercock, Keren Lasker, Yuliya Gordiyenko, Daniel Russel, Andrej Sali, and Carol V. Robinson. Topological models of heteromeric protein assemblies from mass spectrometry: Application to the yeast eIF3:eIF5 complex. *Chemistry & Biology*, 22(1):117–128, **2015**.

[91] Argyris Politis and Antoni J. Borysik. Assembling the pieces of macromolecular complexes: Hybrid structural biology approaches. *PROTEOMICS*, 15(16):2792–2803, **2015**.

[92] Daniel Russel, Keren Lasker, Ben Webb, Javier Velázquez-Muriel, Elina Tjioe, Dina Schneidman-Duhovny, Bret Peterson, and Andrej Sali. Putting the pieces together:

Integrative modeling platform software for structure determination of macromolecular assemblies. *PLoS Biology*, 10(1):e1001244, **2012**.

[93] Weiman Xing, Luca Busino, Thomas R. Hinds, Samuel T. Marionni, Nabiha H. Saifee, Matthew F. Bush, Michele Pagano, and Ning Zheng. SCF^{FBXL3} ubiquitin ligase targets cryptochromes at their cofactor pocket. *Nature*, 496(7443):64–68, **2013**.

[94] Michael R. Hoopmann, Alex Zelter, Richard S. Johnson, Michael Riffle, Michael J. MacCoss, Trisha N. Davis, and Robert L. Moritz. Kojak: Efficient analysis of chemically cross-linked protein complexes. *Journal of Proteome Research*, 14(5):2190–2198, **2015**.

[95] F. Peréz and B.E. Granger. IPython: A system for interactive scientific computing. *Computing in Science Engineering*, 9(3):21–29, **2007**.

[96] Helen Shen. Interactive notebooks: Sharing the code. *Nature*, 515(7525):151–152, **2014**.

[97] Anthony R. Cashmore. Cryptochromes: Enabling plants and animals to determine circadian time. *Cell*, 114(5):537–543, **2003**.

[98] Inês Chaves, Richard Pokorny, Martin Byrdin, Nathalie Hoang, Thorsten Ritz, Klaus Brettel, Lars-Oliver Essen, Gijsbertus T. J. van der Horst, Alfred Batschauer, and Margaret Ahmad. The cryptochromes: Blue light photoreceptors in plants and animals. *Annual Review of Plant Biology*, 62(1):335–364, **2011**.

[99] Chentao Lin and Dror Shalitin. Cryptochrome structure and signal transduction. *Annual Review of Plant Biology*, 54(1):469–496, **2003**.

[100] Gad Asher and Ueli Schibler. Crosstalk between components of circadian and metabolic cycles in mammals. *Cell Metabolism*, 13(2):125–137, **2011**.

[101] Joseph Bass. Circadian topology of metabolism. *Nature*, 491(7424):348–356, **2012**.

[102] Francis Levi and Ueli Schibler. Circadian rhythms: Mechanisms and therapeutic implications. *Annual Review of Pharmacology and Toxicology*, 47(1):593–628, **2007**.

[103] Scott M. Grundy. Metabolic syndrome pandemic. *Arteriosclerosis, Thrombosis, and Vascular Biology*, 28(4):629–636, **2008**.

[104] Susan van Dieren, Joline W.J. Beulens, Yvonne T. van der Schouw, Diederick E. Grobbee, and Bruce Nealb. The global burden of diabetes and its complications: an emerging pandemic. *European Journal of Cardiovascular Prevention & Rehabilitation*, 17(1 Suppl):s3–s8, **2010**.

[105] Paul Zimmet, Dianna Magliano, Yuji Matsuzawa, George Alberti, and Jonathan Shaw. The metabolic syndrome: A global public health problem and a new definition. *Journal of Atherosclerosis and Thrombosis*, 12(6):295–300, **2005**.

[106] Eleonore Maury, Kathryn Moynihan Ramsey, and Joseph Bass. Circadian rhythms and metabolic syndrome: From experimental genetics to human disease. *Circulation Research*, 106(3):447–462, **2010**.

[107] Anders Knutsson. Health disorders of shift workers. *Occupational Medicine*, 53(2):103–108, **2003**.

[108] Biliana Marcheva, Kathryn Moynihan Ramsey, Ethan D. Buhr, Yumiko Kobayashi, Hong Su, Caroline H. Ko, Ganka Ivanova, Chiaki Omura, Shelley Mo, Martha H. Vitaterna, James P. Lopez, Louis H. Philipson, Christopher A. Bradfield, Seth D. Crosby, Lellean JeBailey, Xiaozhong Wang, Joseph S. Takahashi, and Joseph Bass. Disruption of the clock components CLOCK and BMAL1 leads to hypoinsulinaemia and diabetes. *Nature*, 466(7306):627–631, **2010**.

[109] Katja A. Lamia, Kai-Florian Storch, and Charles J. Weitz. Physiological significance of a peripheral tissue circadian clock. *Proceedings of the National Academy of Sciences*, 105(39):15172–15177, **2008**.

[110] Frank A.J.L. Scheer, Michael F. Hilton, Christos S. Mantzoros, and Steven A. Shea. Adverse metabolic and cardiovascular consequences of circadian misalignment. *Proceedings of the National Academy of Sciences*, **2009**.

[111] Steven M. Reppert and David R. Weaver. Molecular analysis of mammalian circadian rhythms. *Annual Review of Physiology*, 63(1):647–676, **2001**.

[112] Chen Liu and Steven M. Reppert. GABA synchronizes clock cells within the suprachiasmatic circadian clock. *Neuron*, 25(1):123–128, **2001**.

[113] Ueli Schibler and Paolo Sassone-Corsi. A web of circadian pacemakers. *Cell*, 111(7):919–922, **2002**.

[114] Carrie L. Partch, Carla B. Green, and Joseph S. Takahashi. Molecular architecture of the mammalian circadian clock. *Trends in Cell Biology*, 24(2):90–99, **2014**.

[115] Charna Dibner, Ueli Schibler, and Urs Albrecht. The mammalian circadian timing system: Organization and coordination of central and peripheral clocks. *Annual Review of Physiology*, 72(1):517–549, **2010**.

[116] Elizabeth S. Maywood, John S. O'Neill, Johanna E. Chesham, and Michael H. Hastings. Minireview: The circadian clockwork of the suprachiasmatic nuclei—analysis of a cellular oscillator that drives endocrine rhythms. *Endocrinology*, 148(12):5624–5634, **2007**.

[117] David K. Welsh, Joseph S. Takahashi, and Steve A. Kay. Suprachiasmatic nucleus: Cell autonomy and network properties. *Annual Review of Physiology*, 72(1):551–577, **2010**.

[118] Steven M. Reppert and David R. Weaver. Coordination of circadian timing in mammals. *Nature*, 418(6901):935–941, **2002**.

[119] Choogon Lee, Jean-Pierre Etchegaray, Felino R.A. Cagampang, Andrew S.I. Loudon, and Steven M. Reppert. Posttranslational mechanisms regulate the mammalian circadian clock. *Cell*, 107(7):855–867, **2001**.

[120] Gijsbertus T. J. van der Horst, Manja Muijtjens, Kumiko Kobayashi, Riya Takano, Shin-ichiro Kanno, Masashi Takao, Jan de Wit, Anton Verkerk, Andre P. M. Eker, Dik van Leenen, Ruud Buijs, Dirk Bootsma, Jan H. J. Hoeijmakers, and Akira Yasui. Mammalian Cry1 and Cry2 are essential for maintenance of circadian rhythms. *Nature*, 398(6728):627–630, **1999**.

[121] Martha Hotz Vitaterna, Christopher P. Selby, Takeshi Todo, Hitoshi Niwa, Carol Thompson, Ethan M. Fruechte, Kenichi Hitomi, Randy J. Thresher, Tomoko Ishikawa, Junichi Miyazaki, Joseph S. Takahashi, and Aziz Sancar. Differential regulation of mammalian *Period* genes and circadian rhythmicity by cryptochromes 1 and 2. *Proceedings of the National Academy of Sciences*, 96(21):12114–12119, **1999**.

[122] Kiho Bae, Xiaowei Jin, Elizabeth S. Maywood, Michael H. Hastings, Steven M. Reppert, and David R. Weaver. Differential functions of *mPer1*, *mPer2*, and *mPer3* in the SCN circadian clock. *Neuron*, 30(2):525–536, **2001**.

[123] Binhai Zheng, Urs Albrecht, Krista Kaasik, Marijke Sage, Weiqin Lu, Sukeshi Vaishnav, Qiu Li, Zhong Sheng Sun, Gregor Eichele, Allan Bradley, and Cheng Chi Lee. Nonredundant roles of the *mPer1* and *mPer2* genes in the mammalian circadian clock. *Cell*, 105(5):683–694, **2001**.

[124] Luca Busino, Florian Bassermann, Alessio Maiolica, Choogon Lee, Patrick M. Nolan, Sofia I.H. Godinho, Giulio F. Draetta, and Michele Pagano. SCF^{Fbxl3} controls the oscillation of the circadian clock by directing the degradation of cryptochrome proteins. *Science*, 316(5826):900–904, **2007**.

[125] Timothy Cardozo and Michele Pagano. The SCF ubiquitin ligase: insights into a molecular machine. *Nature Reviews Molecular Cell Biology*, 5(9):739–751, **2004**.

[126] Sofia I.H. Godinho, Elizabeth S. Maywood, Linda Shaw, Valter Tucci, Alun R. Barnard, Luca Busino, Michele Pagano, Rachel Kendall, Mohamed M. Quwailid, M. Rosario Romero, John O'Neill, Johanna E. Chesham, Debra Brooker, Zuzanna Lalanne, Michael H. Hastings, and Patrick M. Nolan. The after-hours mutant reveals a role for Fbxl3 in determining mammalian circadian period. *Science*, 316(5826):897–900, **2007**.

[127] Sandra M. Siepka, Seung-Hee Yoo, Junghea Park, Weimin Song, Vivek Kumar, Yinin Hu, Choogon Lee, and Joseph S. Takahashi. Circadian mutant *Overtime* reveals F-box protein FBXL3 regulation of *Cryptochrome* and *Period* gene expression. *Cell*, 129(5):1011–1023, **2007**.

[128] Melanie J. Maul, Thomas R.M. Barends, Andreas F. Glas, Max J. Cryle, Tatiana Domratcheva, Sabine Schneider, Ilme Schlichting, and Thomas Carell. Crystal structure and mechanism of a DNA (6-4) photolyase. *Angewandte Chemie International Edition*, 47(52):10076–10080, **2008**.

[129] Edmund A. Griffin Jr., David Staknis, and Charles J. Weitz. Light-independent role of CRY1 and CRY2 in the mammalian circadian clock. *Science*, 286(5440):768–771, **1999**.

[130] An Pan, Eva S. Schernhammer, Qi Sun, and Frank B. Hu. Rotating night shift work and risk of type 2 diabetes: Two prospective cohort studies in women. *PLoS Med*, 8(12):e1001141, **2011**.

[131] Michael H. Hastings, Akhilesh B. Reddy, and Elizabeth S. Maywood. A clockwork web: circadian timing in brain and periphery, in health and disease. *Nature Reviews Neuroscience*, 4(8):649–661, **2003**.

[132] Monica Gallego and David M. Virshup. Post-translational modifications regulate the ticking of the circadian clock. *Nature Reviews Molecular Cell Biology*, 8(2):139–148, **2007**.

[133] Hitoshi Okamura. Clock genes in cell clocks: Roles, actions, and mysteries. *Journal of Biological Rhythms*, 19(5):388–399, **2004**.

[134] Kazuhiro Yagita, Filippo Tamanini, Maya Yasuda, Jan H. J. Hoeijmakers, Gijsbertus T. J. van der Horst, and Hitoshi Okamura. Nucleocytoplasmic shuttling and mCRY-dependent inhibition of ubiquitylation of the mPER2 clock protein. *The EMBO Journal*, 21(6):1301–1314, **2002**.

[135] Michal Sharon and Carol V. Robinson. The role of mass spectrometry in structure elucidation of dynamic protein complexes. *Annual Review of Biochemistry*, 76(1):167–193, **2007**.

[136] Charlotte Uetrecht, Rebecca J. Rose, Esther van Duijn, Kristina Lorenzen, and Albert J.R. Heck. Ion mobility mass spectrometry of proteins and protein assemblies. *Chemical Society Reviews*, 39:1633–1655, **2010**.

[137] William F. Siems, Larry A. Viehland, and Herbert H. Hill Jr. Improved momentum-transfer theory for ion mobility. 1. derivation of the fundamental equation. *Analytical Chemistry*, 84(22):9782–9791, **2012**.

[138] Argyris Politis, Ah Young Park, Suk-Joon Hyung, Daniel Barsky, Brandon T. Ruotolo, and Carol V. Robinson. Integrating ion mobility mass spectrometry with molecular modelling to determine the architecture of multiprotein complexes. *PLoS ONE*, 5(8):e12080, **2010**.

[139] Carol V. Robinson, Andrej Sali, and Wolfgang Baumeister. The molecular sociology of the cell. *Nature*, 450(7172):973–982, **2007**.

[140] Florian Stengel, Andrew J. Baldwin, Matthew F. Bush, Gillian R. Hilton, Hadi Lioe, Eman Basha, Nomalie Jaya, Elizabeth Vierling, and Justin L.P. Benesch. Dissecting heterogeneous molecular chaperone complexes using a mass spectrum deconvolution approach. *Chemistry & Biology*, 19(5):599–607, **2012**.

[141] Keren Lasker, Friedrich Förster, Stefan Bohn, Thomas Walzthoeni, Elizabeth Villa, Pia Unverdorben, Florian Beck, Ruedi Aebersold, Andrej Sali, and Wolfgang Baumeister. Molecular architecture of the 26S proteasome holocomplex determined by an integrative approach. *Proceedings of the National Academy of Sciences*, 109(5):1380–1387, **2012**.

[142] Konstantinos Thalassinos, Arun Prasad Pandurangan, Min Xu, Frank Alber, and Maya Topf. Conformational states of macromolecular assemblies explored by integrative structure calculation. *Structure*, 21(9):1500–1508, **2013**.

[143] Ugo I. Ekeowa, Joanna Freeke, Elena Miranda, Bibek Gooptu, Matthew F. Bush, Juan Pérez, Jeff Teckman, Carol V. Robinson, and David A. Lomas. Defining the mechanism of polymerization in the serpinopathies. *Proceedings of the National Academy of Sciences*, 107(40):17146–17151, **2010**.

[144] Samuel J. Allen, Alicia M. Schwartz, and Matthew F. Bush. Effects of polarity on the structures and charge states of native-like proteins and protein complexes in the gas phase. *Analytical Chemistry*, 85(24):12055–12061, **2013**.

[145] Travis E. Oliphant. Python for scientific computing. *Computing in Science Engineering*, 9(3):10–20, **2007**.

[146] S. van der Walt, S.C. Colbert, and G. Varoquaux. The NumPy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13(2):22–30, **2011**.

[147] Pearu Peterson. F2PY: a tool for connecting Fortran and Python programs. *International Journal of Computational Science and Engineering*, 4(4):296–305, **2009**.

[148] Igor A. Kaltashov and Anirban Mohimen. Estimates of protein surface areas in solution by electrospray ionization mass spectrometry. *Analytical Chemistry*, 77(16):5370–5379, **2005**.

[149] Lorenzo Testa, Stefania Brocca, and Rita Grandori. Charge-surface correlation in electrospray ionization of folded and unfolded proteins. *Analytical Chemistry*, 83(17):6459–6463, **2011**.

[150] David van der Spoel, Erik G. Marklund, Daniel S.D. Larsson, and Carl Caleman. Proteins, lipids, and water in the gas phase. *Macromolecular Bioscience*, 11(1):50–59, **2011**.

[151] Brian C. Bohrer, Samuel I. Merenbloom, Stormy L. Koeniger, Amy E. Hilderbrand, and David E. Clemmer. Biomolecule analysis by ion mobility spectrometry. *Annual Review of Analytical Chemistry*, 1(1):293–327, **2008**.

[152] Michael J. MacCoss. *IDCalc – Isotope Distribution Calculator*. Department of Genome Sciences, University of Washington, Seattle, WA, **2008**.

[153] Hugo Kubinyi. Calculation of isotope distributions in mass spectrometry. a trivial solution for a non-trivial problem. *Analytica Chimica Acta*, 247(1):107–119, **1991**.

[154] Elena N. Kitova, Amr El-Hawiet, Paul D. Schnier, and John S. Klassen. Reliable determinations of protein–ligand interactions by direct ESI-MS measurements. are we there yet? *Journal of The American Society for Mass Spectrometry*, 23(3):431–441, **2012**.

[155] Sibsankar Kundu, Dan C. Sorensen, and George N. Phillips. Automatic domain decomposition of proteins by a Gaussian network model. *Proteins: Structure, Function, and Bioinformatics*, 57(4):725–733, **2004**.

[156] Konstantin V. Korotkov, Maria Sandkvist, and Wim G. J. Hol. The type II secretion system: biogenesis, molecular architecture and mechanism. *Nature Reviews Microbiology*, 10(5):336–351, **2012**.

[157] Maria Sandkvist, Jerry M. Keith, Michael Bagdasarian, and S. Peter Howard. Two regions of EpsL involved in species-specific protein-protein interactions with EpsE and EpsM of the general secretion pathway in *Vibrio cholerae*. *Journal of Bacteriology*, 182(3):742–748, **2000**.

[158] Maria Sandkvist, Michael Bagdasarian, S. Peter Howard, and Victor J. DiRita. Interaction between the autokinase EpsE and EpsL in the cytoplasmic membrane is required for extracellular secretion in *Vibrio cholerae*. *The EMBO Journal*, 14(8):1664–1673, **1995**.

[159] Mark A. Robien, Brian E. Krumm, Maria Sandkvist, and Wim G.J. Hol. Crystal structure of the extracellular protein secretion NTPase EpsE of *Vibrio cholerae*. *Journal of Molecular Biology*, 333(3):657–674, **2003**.

[160] Jodi L. Camberg and Maria Sandkvist. Molecular analysis of the *Vibrio cholerae* type II secretion ATPase EpsE. *Journal of Bacteriology*, 187(1):249–256, **2005**.

[161] Marcella Patrick, Konstantin V. Korotkov, Wim G.J. Hol, and Maria Sandkvist. Oligomerization of EpsE coordinates residues from multiple subunits to facilitate ATPase activity. *Journal of Biological Chemistry*, 286(12):10378–10386, **2011**.

[162] Atsushi Yamagata and John A. Tainer. Hexameric structures of the archaeal secretion ATPase GspE and implications for a universal secretion mechanism. *The EMBO Journal*, 26(3):878–890, **2007**.

[163] Sophia Reindl, Abhrajyoti Ghosh, Gareth J. Williams, Kerstin Lassak, Tomasz Neiner, Anna-Lena Henche, Sonja-Verena Albers, and John A. Tainer. Insights into FlaI functions in archaeal motor assembly and motility from structures, conformations, and genetics. *Molecular Cell*, 49(6):1069–1082, **2013**.

[164] Ana M. Misic, Kenneth A. Satyshur, and Katrina T. Forest. *P. aeruginosa* PilT structures with and without nucleotide reveal a dynamic type IV pilus retraction motor. *Journal of Molecular Biology*, 400(5):1011–1021, **2010**.

[165] Kenneth A. Satyshur, Gregory A. Worzalla, Lorraine S. Meyer, Erin K. Heiniger, Kelly G. Aukema, Ana M. Misic, and Katrina T. Forest. Crystal structures of the pilus retraction motor PilT suggest large domain movements and subunit cooperation drive motility. *Structure*, 15(3):363–376, **2007**.

[166] Joseph D. Mougous, Marianne E. Cuff, Stefan Raunser, Aimee Shen, Min Zhou, Casey A. Gifford, Andrew L. Goodman, Grazyna Joachimiak, Claudia L. Ordoñez, Stephen Lory, Thomas Walz, Andrzej Joachimiak, and John J. Mekalanos. A virulence locus of *Pseudomonas aeruginosa* encodes a protein secretion apparatus. *Science*, 312(5779):1526–1530, **2006**.

[167] Connie Lu, Stewart Turley, Samuel T. Marionni, Young-Jun Park, Kelly K. Lee, Marcella Patrick, Ripal Shah, Maria Sandkvist, Matthew F. Bush, and Wim G.J. Hol. Hexamers of the type II secretion ATPase GspE from *Vibrio cholerae* with increased ATPase activity. *Structure*, 21(9):1707–1717, **2013**.

[168] Julien Marcoux and Carol V. Robinson. Twenty years of gas phase structural biology. *Structure*, 21(9):1541–1550, **2013**.

[169] Adam R. McKay, Brandon T. Ruotolo, Leopold L. Ilag, and Carol V. Robinson. Mass measurements of increased accuracy resolve heterogeneous populations of intact ribosomes. *Journal of the American Chemical Society*, 128(35):11433–11442, **2006**.

[170] Jonathan P. Williams, Jeffery M. Brown, Iain Campuzano, and Peter J. Sadler. Identifying drug metallation sites on peptides using electron transfer dissociation (ETD), collision induced dissociation (CID) and ion mobility-mass spectrometry (IM-MS). *Chemical Communications*, 46:5458–5460, **2010**.

[171] Kenneth J. Laszlo and Matthew F. Bush. Analysis of native-like proteins and protein complexes using cation to anion proton transfer reactions (CAPTR). *Journal of The American Society for Mass Spectrometry*, 26(12):2152–2161, **2015**.

[172] James L. Stephenson Jr. and Scott A. McLuckey. Ion/ion proton transfer reactions for protein mixture analysis. *Analytical Chemistry*, 68(22):4026–4032, **1996**.

[173] Bruce B. Reinhold and Vernon N. Reinhold. Electrospray ionization mass spectrometry: Deconvolution by an entropy-based algorithm. *Journal of the American Society for Mass Spectrometry*, 3(3):207–215, **1992**.

[174] A.G. Ferrige, M.J. Seddon, B.N. Green, S.A. Jarvis, J. Skilling, and J. Staunton. Disentangling electrospray spectra with maximum entropy. *Rapid Communications in Mass Spectrometry*, 6(11):707–711, **1992**.

[175] Nina Morgner and Carol V. Robinson. Massign: An assignment strategy for maximizing information from the mass spectra of heterogeneous protein assemblies. *Analytical Chemistry*, 84(6):2939–2948, **2012**.

[176] Yao-Hsin Tseng, Charlotte Uetrecht, Shih-Chieh Yang, Arjan Barendregt, Albert J. R. Heck, and Wen-Ping Peng. Game-theory-based search engine to automate the mass assignment in complex native electrospray mass spectra. *Analytical Chemistry*, 85(23):11275–11283, **2013**.

[177] J. Andrew Aquilina, Justin L. P. Benesch, Orval A. Bateman, Christine Slingsby, and Carol V. Robinson. Polydispersity of a mammalian chaperone: Mass spectrometry reveals the population of oligomers in αB-crystallin. *Proceedings of the National Academy of Sciences*, 100(19):10611–10616, **2003**.

[178] Andrew J. Baldwin, Hadi Lioe, Carol V. Robinson, Lewis E. Kay, and Justin L.P. Benesch. αB-crystallin polydispersity is a consequence of unbiased quaternary dynamics. *Journal of Molecular Biology*, 413(2):297–309, **2011**.

[179] K. Jarrod Millman and Michael Aivazis. Python for scientists and engineers. *Computing in Science Engineering*, 13(2):9–12, **2011**.

[180] J.D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science Engineering*, 9(3):90–95, **2007**.

[181] Sibsankar Kundu, Julia S. Melton, Dan C. Sorensen, and George N. Phillips Jr. Dynamics of proteins in crystals: Comparison of experiment with simple models. *Biophysical Journal*, 83(2):723–732, **2002**.

[182] G. Kirchhoff. Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer Ströme geführt wird. *Annalen der Physik*, 148(12):497–508, **1847**.

[183] D. Babić, D.J. Klein, I. Lukovits, S. Nikolić, and N. Trinajstić. Resistance-distance matrix: A computational algorithm and its application. *International Journal of Quantum Chemistry*, 90(1):166–176, **2002**.

[184] Bojan Mohar. The Laplacian spectrum of graphs. In Yousef Alavi, G. Chartrand, O.R. Oellermann, and A.J. Schwenk (Editors), *Graph Theory, Combinatorics, and Applications*, volume 2, pages 871–898. Wiley, **1991**.

[185] Glenn Murray. Rotation about an arbitrary axis in 3 dimensions. `url: http://inside.mines.edu/fs_home/gmurray/ArbitraryAxisRotation/`, **2013**.

Appendix A

DOMAIN DECOMPOSITION OF CRY2–FBXL3–SKP1 WITH A GAUSSIAN NETWORK MODEL (GNM)

A.1 Introduction

Visual identification of mobile domains in proteins and protein complexes is seldom reproducible as 2D visualization is insufficient for gauging the flexibility and connectivity of 3D structures. Conversely, methods for describing the physical nature of proteins in an automatic and analytical way take the judgment away from the researcher and allow conclusions to be drawn free from bias, while providing quantitative data about the mobility of large and complex systems. Kundu et al. have described a method of using an elastic network model (ENM) or Gaussian network model (GNM) [181] to automatically decompose mobile domains of monomeric proteins [155]. We have extended this to the study of multisubunit protein complexes — in this case CRY2–FBXL3–SKP1 — by performing the analysis on each subunit as a monomer and visually confirming that the mobility of the identified domains are not hindered by their neighbors.

The approach will be described over the next several sections, first applying the algorithm to a simple system that is easily deconvoluted visually and then extending the algorithm to CRY2–FBXL3–SKP1. The software is initialized by importing the following libraries.

```
In [1]: import numpy as np
        import pandas as pd
        import scipy.spatial.distance as dist
        from matplotlib import pyplot as plt
        %matplotlib inline
```

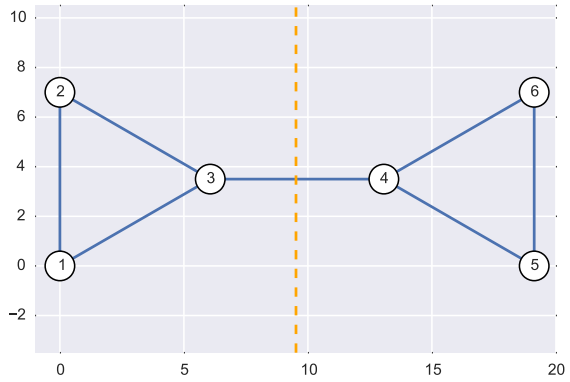
A.2 GNM Example — A Simple 6-node Graph

To borrow the rather elegant example provided by Kundu et al. [155] for explaining the algorithm, let us first consider a simple system that can be represented as a graph of 6 nodes. Edges are drawn in to signify adjacency, although the algorithm itself will define adjacency based on Euclidean distance. In this example, the graph is drawn such that connected nodes will be spaced 7.0 arbitrary units apart.

```
In [2]: # Set the distance between nodes
        d = 7.0

        # For points connected diagonally (an equilateral triangle), how much should
        # the x-value change
        x = np.sin(np.pi/3) * d

        graph = np.array([[    0.0, 0.0],
                          [      0,   d],
                          [      x, d/2],
                          [  x + d, d/2],
                          [2*x + d, 0.0],
                          [2*x + d,   d]])
```

```
In [3]: plt.scatter(*graph.T, marker='o', facecolor='white', s=360)
        plt.xlim(-1,20)
        plt.ylim(-3.5,10.5)
        [plt.text(g[0],g[1], str(i+1)) for i,g in enumerate(graph)]
        plt.vlines((x + d/2), -3.5, 10.5, color='orange', linestyles='--')
```

```
Out[3]: <matplotlib.collections.LineCollection at 0x115174f90>
```



The Laplacian matrix (L), or Kirchhoff matrix [182,183], describes both the valency and the connectivity of each node as given by the logic shown in Equation A.1. It is a square, symmetric matrix with the dimensions equal to the total number of nodes. Each unit of the matrix l_{ij} represents the relationship between the two corresponding nodes i and j, with a -1 representing adjacency except for the diagonal of the matrix, where $i = j$. Cells along the diagonal describe the valency of each node — how many nodes are adjacent. In the case of a protein, adjacency of residues is approximated by representing the α-carbons as nodes and determining adjacency from the distance between α-carbons. Given two α-carbons i and j, if the distance R_{ij} between them is less than or equal to some critical radius r_c, then they are adjacent.

$$l_{i,j} = \begin{cases} -1 & \text{if } i \neq j \text{ and } R_{ij} \leq r_c \\ 0 & \text{if } i \neq j \text{ and } R_{ij} > r_c \\ -\sum\limits_{i,j \neq j} l_{i,j} & \text{if } i = j \end{cases} \qquad L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix} \tag{A.1}$$

Specifically, the Laplacian matrix is given by subtracting the adjacency matrix (A) from the degree matrix (D), shown in Equation A.2. The adjacency matrix describes the edges in the graph, such that if if vertex i and vertex j are adjacent, then both a_{ij} and a_{ji} will be equal to 1, and otherwise equal to 0. The degree matrix shows the valency of each node along the diagonal, such that if $i = j$ then d_{ij} will be the number of edges connecting to that node. The diagonal of the matrix is equivalent to the vector that would result from summing either the rows or columns of the adjacency matrix.

$$L = D - A = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \tag{A.2}$$

The algorithm can be implemented in Python as follows. The graph has been defined such that connected nodes have a distance (variable `d`) of 7.0, and consequently connection can be defined by inter-node distance merely by using an r_c of 7.0. First, a distance matrix of the graph is generated from the array `graph` of coordinates.

```
In [4]: distances = dist.cdist(graph,graph).round()
```

```
Out[4]: array([[  0.,   7.,   7.,  14.,  19.,  20.],
               [  7.,   0.,   7.,  14.,  20.,  19.],
               [  7.,   7.,   0.,   7.,  14.,  14.],
               [ 14.,  14.,   7.,   0.,   7.,   7.],
               [ 19.,  20.,  14.,   7.,   0.,   7.],
               [ 20.,  19.,  14.,   7.,   7.,   0.]])
```

The adjaceny matrix should represent the connected nodes ($R_{ij} \leq r_c$) with a 1. Because nodes by definition have a distance of 0.0 against themselves, we must exclude values of 0.0 to prevent the diagonal from having a value of 1, thus the logic of defining adjacency is when distance is $0 < R_{ij} \leq r_c$.

```
In [5]: adjacency = ((distances > 0) & (distances <= 7.0)).astype('int')
```

```
Out[5]: array([[0, 1, 1, 0, 0, 0],
               [1, 0, 1, 0, 0, 0],
               [1, 1, 0, 1, 0, 0],
               [0, 0, 1, 0, 1, 1],
               [0, 0, 0, 1, 0, 1],
               [0, 0, 0, 1, 1, 0]])
```

The programmatically generated adjacency matrix is equivalent to the previously described matrix that was assigned visually. As the degree matrix (D) is defined as the number of edges connecting the node, the adjacency matrix can be summed along either the horizontal or vertical axis to get a vector of the valency. To format this along the diagonal, this vector can be multiplied element-wise by the identity matrix, yielding the degree matrix.

```
In [6]: valency = (adjacency.sum(axis=0) * np.identity(graph.shape[0])).astype('int')
```

```
Out[6]: array([[2, 0, 0, 0, 0, 0],
               [0, 2, 0, 0, 0, 0],
               [0, 0, 3, 0, 0, 0],
               [0, 0, 0, 3, 0, 0],
               [0, 0, 0, 0, 2, 0],
               [0, 0, 0, 0, 0, 2]])
```

Finally, subtraction is performed to generate the Laplacian matrix (L).

```
In [7]: valency - adjacency
```

```
Out[7]: array([[ 2, -1, -1,  0,  0,  0],
               [-1,  2, -1,  0,  0,  0],
               [-1, -1,  3, -1,  0,  0],
               [ 0,  0, -1,  3, -1, -1],
               [ 0,  0,  0, -1,  2, -1],
               [ 0,  0,  0, -1, -1,  2]])
```

The result is the same matrix that was constructed visually. These steps can be assembled into a function for reuse when the protein monomers are analyzed.

```
In [8]: def Laplacian(coords, rc):
            distances = dist.cdist(coords,coords)
            adjacency = ((distances > 0) & (distances <= rc))
            valency = adjacency.sum(axis=0) * np.identity(coords.shape[0])
            return (valency - adjacency).astype('int')
```

```
In [9]: Laplacian(graph, 7.0)
```

```
Out[9]: array([[ 2, -1, -1,  0,  0,  0],
               [-1,  2, -1,  0,  0,  0],
               [-1, -1,  3, -1,  0,  0],
               [ 0,  0, -1,  3, -1, -1],
               [ 0,  0,  0, -1,  2, -1],
               [ 0,  0,  0, -1, -1,  2]])
```

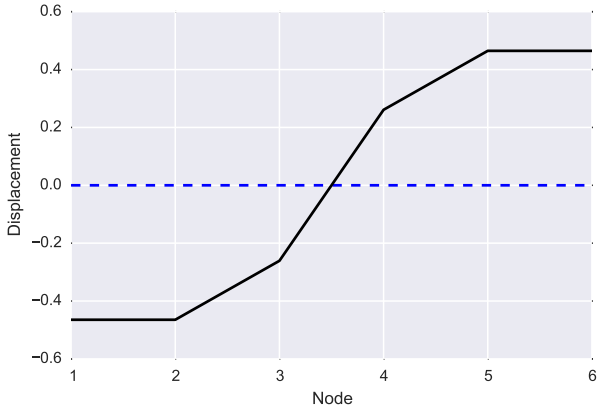
The next step is to perform singular value decomposition (SVD) on the Laplacian matrix. The function as implemented in NumPy (`numpy.linalg.svd`) returns a vector of eigenvalues `s` and the corresponding array of eigenvectors `V`. The second smallest eigenvalue is called the *algebraic connectivity* and this value represents the connectedness of the graph, with nonzero values indicating that a graph is in fact connected [184]. Areas of the graph where connectedness is diminished indicate increased flexibility. More intuitively, one can imagine that if α-carbons are in close proximity, they would experience more interaction, and it is less likely that the protein would be flexible at that point. The eigenvector corresponding to the algebraic connectivity is called the Fielder vector. Points where the Fiedler vector intersects the x-axis indicate lower connectivity and thus greater flexibility in the graph. In the case of a graph representing a protein, this is an interface between two mobile domains. To extract the Fielder vector programmatically, a copy is made of the array `s` and is then sorted. The second value in the sorted array `s_sort[1]` is the eigenvalue that corresponds to the Fiedler vector. Thus, the position of this value in the original eigenvalue array `s` gives the position of the Fiedler vector in the array `V`.

```
In [10]: U, s, V = np.linalg.svd(L, full_matrices=True)

In [11]: s_sort = s.copy()
         s_sort.sort()
         Fiedler = V[s == s_sort[1]][0]

Out[11]: array([-0.46470513, -0.46470513, -0.26095647,  0.26095647,  0.46470513,
                0.46470513])
```

The SVD step should also be packaged as a function for reuse throughout the remainder of this analysis.

```
In [12]: def get_Fiedler(L):
             U, s, V = np.linalg.svd(L, full_matrices=True)
             s_sort = s.copy()
             s_sort.sort()
             return V[s == s_sort[1]][0]

In [13]: get_Fiedler(L)

Out[13]: array([-0.46470513, -0.46470513, -0.26095647,  0.26095647,  0.46470513,
                 0.46470513])
```

```
In [14]: plt.plot(np.arange(1,7), Fiedler, '-', color = 'black')
         plt.hlines(0,1,6, color='blue', linestyle='--')
         plt.xlabel('Node')
         plt.ylabel('Displacement')

Out[14]: <matplotlib.lines.Line2D at 0x10f856f50>
```



As we'd expect from visually inspecting the graph, the Fiedler vector crosses the x-axis between nodes 3 and 4, indicating that this is the most flexible point in the graph. Using this simple example borrowed from Kundu et al. [155], the utility of this algorithm is demonstrated. Further, it is clear that the algorithm has been successfully implemented in Python and can be reused for the remainder of this research.

A.3 Application of the GNM to CRY2–FBXL3–SKP1

This algorithm was originally proposed for use on monomeric proteins, and the authors provided no examples of it being validated for multiprotein complexes. It was therefore our intention to test the algorithm on CRY2–FBXL3–SKP1 and evaluate the results of domain decomposition. Treating the entire complex as a single chain is problematic, however. Each subunit is a separate chain, and as such the distance between the C-terminus of one subunit and the N-terminus of the next is much farther than the typical space between residues in sequence. It should therefore be preferable to represent each chain as a separate graph, otherwise one would expect to see false negatives in between subunit chains where connectivity would appear to be very low. This would manifest as the Fiedler vector abruptly

changing sign. To demonstrate this issue, the algorithm will first be demonstrated on the entire complex as a single chain and then on each subunit individually.

The following function loads the PDB file as a data.

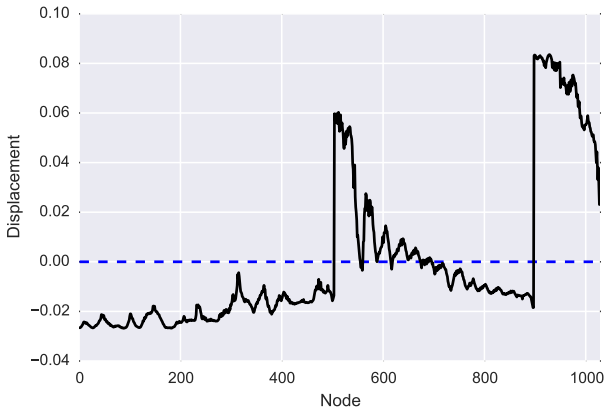
```
In [15]: def pdb_to_df(fname):
             with open(fname, 'r') as PDBfile:
                 PDBlines = PDBfile.read().splitlines()
                 PDBlines = filter(lambda x: x[0:4] == 'ATOM', PDBlines)

             ATOMdata = [[x[0:6],
                          x[6:11],
                          x[12:16],
                          x[16:17],
                          x[17:20],
                          x[21:22],
                          x[22:26],
                          x[26:27],
                          x[30:38],
                          x[38:46],
                          x[46:54],
                          x[54:60],
                          x[60:66],
                          x[76:78],
                          x[78:80]] for x in PDBlines]

             headers = ['Record', 'serial', 'name', 'altLoc', 'Resn',
                     'chainID', 'resSeq', 'iCode', 'x', 'y', 'z', 'occupancy',
                     'tempFactor', 'element', 'charge']

             ATOMdata = [[s.strip() for s in inner] for inner in ATOMdata]
             df = pd.DataFrame(ATOMdata, columns=headers)
             df = df.convert_objects(convert_numeric=True)
             df = df.set_index(df['serial'].values)

             return df

In [16]: atoms = pdb_to_df('4i6j.pdb')
```

Next, the Cartesian coordinates of the α-carbons are subset from the data frame as a single chain and processed using the domain decomposition algorithm.

```
In [17]: entire_complex = atoms[(atoms['name'] == 'CA')][['x', 'y', 'z']]
         L = Laplacian(A, rc=10.0)
         Fiedler = get_Fiedler(L)
```

```
In [18]: m, n = entire_complex.shape

         plt.plot(np.arange(m)+1, Fiedler, color = 'black')
         plt.hlines(0,1,m+1, color='blue', linestyle='--')
         plt.xlim(0, m+1)
         plt.xlabel('Node')
         plt.ylabel('Displacement')
```

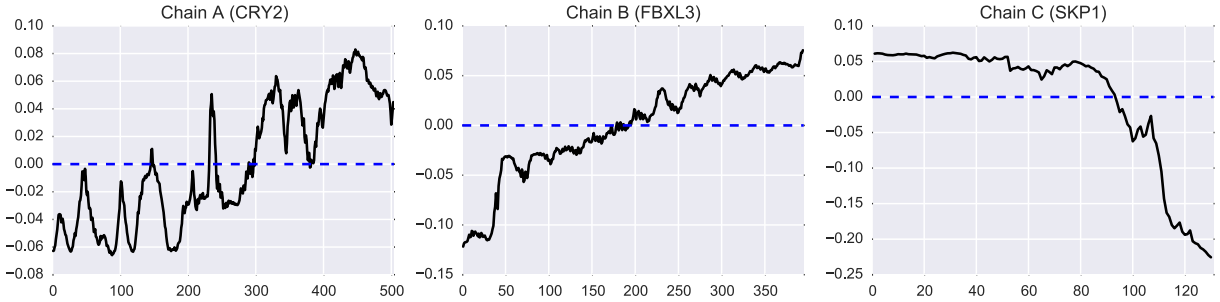
```
Out[18]: <matplotlib.lines.Line2D at 0x110071510>
```



Sharp vertical jumps are visible between nodes 503 and 504 (the CRY2 C-terminus and the FBXL3 N-terminus), as well as between nodes 897 and 898 (FBXL3 C-terminus and the SKP1 N-terminus). The mobile domains cannot be reliably predicted in this manner. The next step is therefore to treat each subunit as a separate graph and perform the analysis on each one individually.

```
In [19]: chains = {'A': 'CRY2', 'B': 'FBXL3', 'C': 'SKP1'}

         for i, chain, in enumerate(['A', 'B', 'C']):
             subunit = atoms[(atoms['name'] == 'CA') &
                             (atoms['chainID'] == chain)][['x', 'y', 'z']].values

             L = Laplacian(subunit, rc=10.0)
             Fiedler = get_Fiedler(L)
             m, n = subunit.shape

             plt.plot(np.arange(m)+1, Fiedler, color = 'black')
             plt.hlines(0,1,m+1, color='blue', linestyle='--')
             plt.xlim(0, m+1)
             plt.title('Chain %s (%s)' % (chain, name))
```

```
Out[19]: <matplotlib.lines.Line2D at 0x110aed8d0>
```



For each subunit, the Fiedler vector intersects the x-axis one or more times. The CRY2 Fiedler vector intersects the x-axis multiple times, however Kundu et al. suggest that segments of the Fielder vector that cross the x-axis but are ten residues or shorter should not be considered mobile domains due to their length not being sufficient to confer significant flexibility. The CRY2(256–265) segment that is positive is right on the cusp of this cutoff, however upon further investigation it becomes clear that this apparent flexibility is due to the absence of CRY2(252–255) in the crystal structure (PDB: 4I6J). The only other residues missing from CRY2 in the crystal structure are CRY2(1–20) and CRY2(528–544), which are the N- and C-terminal residues and therefore do not cause a break in the sequence of nodes that are subjected to analysis. For FBXL3, there is also substantial fluctuation at the x-axis between nodes 179–191, which correspond to FBXL3(213–225). Visualization of the structure reveals that the residues are part of an α-helix, with FBXL3(226) being the beginning of a weak point in the bent β-sheet. The SKP1 Fiedler vector crosses the x-axis only once with a very clean drop between nodes 93–94, which correspond to SKP1(125–126). The missing residues for this subunit include SKP1(1–2, 34–43, 65–84, 163) — quite a substantial number of residues breaking the sequence, however upon visual inspection of the structure, these are not close in space to the domain interface, and therefore cannot account for the change in sign of the Fiedler vector.

```
In [20]: interfaces = {'A': 297, 'B': 191, 'C': 92}
```

Although visual inspection confirmed that none of the missing residues created a false interface between domains, it is still possible that missing domains may shift the Fiedler

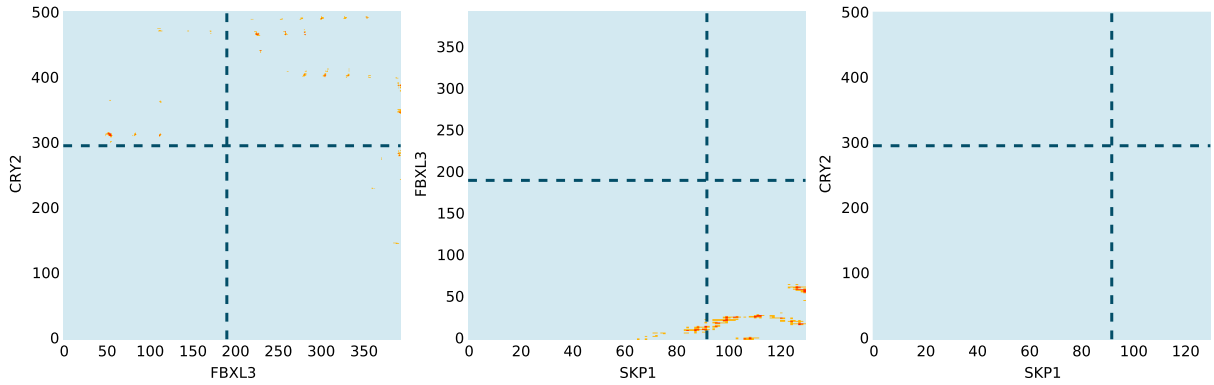
vector by a small amount, so manual adjustment of a small number of residues was performed to ensure that rigid structures such as α-helices would not be broken. Therefore, three domain interfaces have been identified for CRY2–FBXL3–SKP1 — CRY2(320:321), FBXL3(223:224), and SKP1(129:130).

A.4 Examining Intersubunit Interactions

Mobile domains were identified in each subunit, however interactions with neighboring subunits must also be considered when determining the actual flexibility at these interfaces. By extending the previously implemented approximation of distance between α-carbons as determining connectivity, we can automatically determine if the flexibility of any giving domain is inhibited by intersubunit interactions.

In the following analysis, domain interfaces are shown by dashed lines. Interactions are based on either an r_c of 7.0 (red) or 10.0 (yellow). Both values of r_c are evaluated to gauge the degree of interaction and the number of interaction between each domain, however whether this interaction restricts independent movement is somewhat subjective.

```
In [21]: cmap = mpl.colors.ListedColormap(['#D3E9F1', '#FFB400', '#FF2800'])

         for i, chain in enumerate([('A','B'), ('B','C'), ('A','C')]):

             M, N = [atoms[(atoms['name'] == 'CA') &
                           (atoms['chainID'] == c)][['x', 'y', 'z']].values for c in chain]

             interaction7 = dist.cdist(M,N) <= 7.0
             interaction10 = dist.cdist(M,N) <= 10.0
             interaction = interaction7.astype('int') + interaction10.astype('int')

             m,n = [x.shape[0] for x in [M,N]]
             plt.pcolormesh(interaction, cmap=cmap)
             plt.xlabel(chains[chain[1]])
             plt.ylabel(chains[chain[0]])
             plt.vlines(interfaces[chain[1]], 0, m, color='#024E68', linestyles='--')
             plt.hlines(interfaces[chain[0]], 0, n, color='#024E68', linestyles='--')
             plt.xlim(0,n)
             plt.ylim(0,m)

Out[21]: <matplotlib.collections.QuadMesh at 0x11d51da90>
```



The N-terminal domain CRY2 shows very little interaction with FBXL3 and absolutely no interaction with SKP1, suggesting that it can in fact move independently of the rest of the complex. The C-terminal domains of FBXL3 and CRY2 show a great deal of interaction, but this is known from inspecting the crystal structure and is obvious visually. Therefore structures generated through bending at the CRY2 pivot point should involve the N-terminal domain moving relative to the rest of the complex.

As mentioned previously, there is a great deal of interaction between the C-terminal domain of FBXL3 and the C-terminal domain of CRY2, however this is known from a more thorough analysis of the crystal structure — the C-terminal tail of FBXL3 is docked in the CRY2 binding pocket and there is substantial interaction between CRY2 and the leucine-rich-repeat (LRR) of FBXL3 [93]. The portion of the LRR that belongs to the N-terminal domain of FBXL3 also interacts with CRY2, however it is clear from the interaction plot that the N-terminal domain is quite free to move relative to the C-terminal domain. As the N-terminal domain of FBXL3 is known to interact with SKP1 both from the crystal structure and from the above interaction studies, it suggests that the pivot point of FBXL3 involves the motion of the N-terminus of FBXL3 moving with SKP1 independently of CRY2.

Finally, there is the motion of SKP1 to consider. SKP1 has no interaction with CRY2 whatsoever. The C-terminal domain of SKP1 shows a great deal of interaction with FBXL3, but once again it is clear that the N-terminal domain — though it does have interactions with FBXL3 — is indeed more free to move in comparison.

A.5 Conclusions

A previously described GNM for identifying mobile domains in monomeric proteins has been successfully implemented in Python. The previously described methodology has been extended to identify mobile domains in multisubunit protein complexes, and was used to identify three potential pivot points in CRY2–FBXL3–SKP1 — one in each subunit. Domain decomposition in each subunit is automatic, followed by visual inspection of the intersubunit distances between α-carbons to verify that interactions with neighboring subunits are not restricting the flexibility of the domain.

Appendix B

A MODEL FOR GENERATING AND CHARACTERIZING SIMULATED STRUCTURES OF CRY2–FBXL3–SKP1

B.1 Introduction

The material presented in this appendix includes the Python code used to generate candidate structures of the extended conformer of CRY2–FBXL3–SKP1 and calculate the collision cross sections, number of steric clashes, and cross-link distances in these simulated structures. These data are then analyzed with a scoring function. The results of this model and the scoring function are analyzed and discussed in depth in Chapter 3.

The script is initialized by importing the following libraries. NumPy (`numpy`) provides an interface for working with C-like arrays, along with an extensive library of linear algebra functions. `pandas` is a library that allows data to be represented as a `DataFrame` object, providing a high-level interface for working with labeled data. SciPy (`scipy`) is a library with built-in functions for various scientific applications. For this model the `cdist` function from the `scipy.spatial.distance` module calculates a Euclidean distance matrix from two arrays of Cartesian coordinates. The `EHSS` module is a compiled FORTRAN module used to calculate collision cross sections and is described in further detail below (Section B.2.3).

```
In [1]: import time
        import numpy as np
        import pandas as pd
        import scipy.spatial.distance as dist
        import EHSS
```

B.2 Definition of functions

B.2.1 `pdb_to_df`*: parsing PDB* `ATOM` *data as a* `pandas.DataFrame` *object*

Protein Data Bank (PDB) files contain a great deal of atomic-level structural information about biomolecules characterized by X-ray crystallography and other high-resolution techniques. For the majority of the modeling performed in this thesis, the most relevant data are those that describe the atoms themeselves. Lines of data such as these begin with `ATOM` and contain identify each atom by a serial number, chain ID, residue name, residue number, etc., as well as the cartesian coordinates of the atom. For tabular data like this, a `pandas.DataFrame` object is perhaps one of the most convenient container for such information, rather than parsing individual columns and storing them in separate arrays. Further, when wrapped as a function it is quite reusable and consequently makes an appearance in all of the author's code that works with PDB files.

```
In [2]: def pdb_to_df(fname):
            with open(fname, 'r') as PDBfile:
                PDBlines = PDBfile.read().splitlines()
                PDBlines = filter(lambda x: x[0:4] == 'ATOM', PDBlines)

            ATOMdata = [[x[0:6], x[6:11], x[12:16], x[16:17], x[17:20], x[21:22],
                         x[22:26], x[26:27], x[30:38], x[38:46], x[46:54], x[54:60],
                         x[60:66], x[76:78], x[78:80]] for x in PDBlines]

            headers = ['Record', 'serial', 'name', 'altLoc', 'Resn',
                       'chainID', 'resSeq', 'iCode', 'x', 'y', 'z', 'occupancy',
                       'tempFactor', 'element', 'charge']

            ATOMdata = [[s.strip() for s in inner] for inner in ATOMdata]
            df = pd.DataFrame(ATOMdata, columns=headers)
            df = df.convert_objects(convert_numeric=True)
            df = df.set_index(df['serial'].values)

            return df
```

B.2.2 `RotatePoint`*: bending the structure at the domain interfaces*

The `RotatePoint` function rotates an array of coordinates about three axes orthogonal axes. The following equation [185] is used to rotate a point (x, y, z) about some axis in 3D space by

angle θ, giving a new point (x', y', z'). The axis is defined by a unit vector (u, v, w) passing through a point (a, b, c)

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} (a(v^2+w^2) - u(bv+cw-ux-vy-wz))(1-\cos\theta) + x\cos\theta + (-cv+bw-wy+vz)\sin\theta \\ (b(u^2+w^2) - v(au+cw-ux-vy-wz))(1-\cos\theta) + y\cos\theta + (cu-aw+wx-uz)\sin\theta \\ (c(u^2+v^2) - w(au+bv-ux-vy-wz))(1-\cos\theta) + z\cos\theta + (-bu+av-vx+uy)\sin\theta \end{bmatrix}$$

This equation is broadly applicable and can be used for any single axis. For our system, we chose to simply use the x-, y-, and z-axes for rotation. The angles about the x-, y-, and z-axes will be called φ, θ, and ψ respectively. As the unit vectors for these axes are simply $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$, respectively, the equations for each can be greatly simplified. For example, rotation about the x-axis simplifies to

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ b + (y-b)\cos\varphi + (c-z)\sin\varphi \\ c + (z-c)\cos\varphi + (y-b)\sin\varphi \end{bmatrix}$$

Rotation about the y- and z-axes are, in turn, respectively given by

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a + (x-a)\cos\theta + (z-c)\sin\theta \\ y \\ c + (z-c)\cos\theta + (a-x)\sin\theta \end{bmatrix} \qquad \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a + (x-a)\cos\psi + (b-y)\sin\psi \\ b + (y-b)\cos\psi + (x-a)\sin\psi \\ z \end{bmatrix}$$

If rotation is always performed in the same order, first about the x-axis by φ, then the y-axis by θ, and finally the z-axis by ψ, substitution can bring this down to a single equation. If a point is first rotated about the x-axis and then the y-axis, the equation for the x-axis is subsituted into the equation for the y-axis as follows

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a + (x-a)\cos\theta + (z-c)\sin\theta\cos\varphi + \sin\theta(y-b)\sin\varphi \\ b + \cos\varphi(y-b) + \sin\varphi(c-z) \\ c + (a-x)\sin\theta + \sin\varphi\cos\theta(z-c) + \cos\theta\cos\varphi(y-b) \end{bmatrix}$$

This equation is then substituted in the equation for rotation about the z-axis to give the following.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a + (x-a)\cos\theta\cos\psi + (y-b)(\sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi) \\ \qquad + (z-c)(\cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi) \\ b + (x-a)\cos\theta\sin\psi + (y-b)(\cos\varphi\cos\psi + \sin\theta\sin\varphi\sin\psi) \\ \qquad + (z-c)(\cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi) \\ c + (a-x)\sin\theta + (y-b)\sin\varphi\cos\theta + (z-c)\cos\varphi\cos\theta \end{bmatrix}$$

which, in a single equation, describes the rotation of a point (x, y, x) — or an array of such points — about the x-, y-,and z-axes by angles φ, θ, and ψ, after the structure has been translated such that the point (a, b, c) is at the origin.

This formula is implemented such that the point (a, b, c) is the hinge-like point between two mobile domains as identified by the Gaussian network model (See Appendix A). At this point, the protein complex is divided into two halves, one of which will remain static while the other rotates about the point by angles φ, θ, and ψ. The mobile half will rotate about these axes in $\frac{\pi}{15}$ rad increments, rotating 2π rad about the x- and y-axes and π rad about the z-axis. This function is implemented in Python as follows

```
In [3]: def RotatePoint(coords, point, phi=0.0, theta=0.0, psi=0.0):
            p = np.zeros(np.shape(coords))
            a,b,c = point
            x = coords[:,0]
            y = coords[:,1]
            z = coords[:,2]

            ct = np.cos(theta)
            st = np.sin(theta)
```

```
cp = np.cos(phi)
sp = np.sin(phi)
cy = np.cos(psi)
sy = np.sin(psi)

p[:,0] = a + (x-a)*ct*cy + (y-b)*(sp*st*cy - cp*sy) + (z-c)*(cp*st*cy + sp*sy)
p[:,1] = b + (x-a)*ct*sy + (y-b)*(cp*cy + sp*st*sy) + (z-c)*(cp*st*sy - sp*cy)
p[:,2] = c + (a-x)*st + (y-b)*sp*ct + (z-c)*ct*cp

return p
```

B.2.3 `CalcCCS`*: collision cross section calculations*

An algorithm for calculating collision cross sections using both the projection approximation (PA) and exact hard-spheres scattering (EHSS) of a structure was written by Alex Shvartsburg and has been described previously in the literature [65]. The code for this algorithm, written in FORTRAN, is under a license not to be distributed and is therefore not included in this appendix. The NumPy package conveniently provides a software tool, F2PY, which compiles FORTRAN code into a Python library, allowing it to be directly interfaced with Python code [147]. The wrapper function is shown below. It calls the external module `EHSS` and sends it the array `data` of coordinates and masses. The second argument to the function is an empty array `dummy`, which stores the results of the calculations. The third argument determines the number of replicate calculations per collision cross section calculation; it has been set to 10 000. `num` is the number of collision cross sections that should be calculated in total, and is used to define the number of rows in the `dummy` array. Finally, `nu` is the number of rows in the incoming `data` array, which is the number of atoms. This allows the FORTRAN module, which requires static type declaration, to declare the array of atoms at the initialization of the calculation. This function returns an array of collision cross sections in an $n \times 2$ array, where n is the number of calculations. The first column contains the collision cross sections calculated using the PA, and the second column contains those calculated using the EHSS. A linear combination of these will be used for the final value, as described in Section 3.3.

```
In [4]: def CalcCCS(data, num):
            dummy = np.zeros((num,2), dtype=float)
            nu = np.shape(data)[0]
            return EHSS.crossrotate(data, dummy, 10000, num, nu)
```

B.2.4 `ResnConvert`*: accounting for sequence differences among data files*

The numbering of the protein sequence in the cross-links data file differs slightly for some of the lysine resides as compared to the published crystal structure (PDB: 4I6J), however it changes throughout the sequence. For this reason, a simple shift of a few characters will not work, so a dictionary is wrapped in a function with an if/else clause. For each entry in the cross-link data file, chain ID and residue number are passed to the function and if the residue is in the dictionary, the sequence number is updated to the correct value. The `ResnConvert` function is more of a convenience function to move this dictionary out of the main body of the code.

```
In [5]: def ResnConvert(pro, resn):
            exceptions = {('B',  22):  20,
                          ('B',  24):  22,
                          ('B',  25):  23,
                          ('B',  94):  92,
                          ('B', 102): 100,
                          ('B', 106): 104,
                          ('B', 118): 116,
                          ('B', 123): 121,
                          ('B', 142): 140,
                          ('B', 174): 172,
                          ('B', 180): 178,
                          ('B', 192): 190,
                          ('B', 203): 201,
                          ('B', 206): 204,
                          ('B', 250): 248,
                          ('B', 404): 402,
                          ('B', 416): 414,
                          ('C',  50):  56,
                          ('C',  51):  57,
                          ('C',  80):  94,
                          ('C', 107): 121,
                          ('C', 114): 128,
                          ('C', 116): 130,
                          ('C', 123): 137,
                          ('C', 128): 142,
```

```
                    ('C', 141): 155,
                    ('C', 149): 163}

        if (pro,resn) in exceptions:
            resn = exceptions[(pro,resn)]

        return resn
```

B.2.5 `read_xlinks`*: parsing cross-link data files*

The `read_xlinks` function reads and parses the cross-link data files to a pandas data frame.

```
In [6]: def read_xlinks(xlinkfiles, prots):
            frames = [pd.read_csv(xlinkfile, delimiter='\t') for xlinkfile in xlinkfiles]
            xlinks = pd.concat(frames, ignore_index=True)

            xlinks = xlinks[xlinks['TYPE'] == 'CROSSLINK']

            headers = ['PROTEIN 1', 'POSITION', 'PROTEIN 2', 'POSITION.1']
            xlinks = xlinks[headers]
            xlinks = xlinks.drop_duplicates()

            # Convert the protein name to it's appropriate chain ID in the PDB file
            # using the 'prots' dictionary.
            xlinks.replace({'PROTEIN 1': prots}, inplace=True)
            xlinks.replace({'PROTEIN 2': prots}, inplace=True)

            # Use the ResnConvert function to change the residue number to the one that
            # agrees with the PDB File
            xlinks2 = xlinks.values.tolist()
            xlinks2 = [[xl[0], ResnConvert(xl[0],xl[1]),
                       xl[2], ResnConvert(xl[2],xl[3])] for xl in xlinks2]


            # Drop cross-links involving residues that are not in the PDB Structure
            invalid = [('A', 2), ('C', 163), ('B', 20), ('B', 323)]
            xlinks2 = filter(lambda d: (tuple(d[0:2]) not in invalid)
                             and (tuple(d[2:4]) not in invalid), xlinks2)

            xlinks = pd.DataFrame(xlinks2, columns=headers)

            return xlinks
```

B.2.6 `EvalXlinks`*: calculating the distance between cross-linked residues*

```
In [7]: def EvalXlinks(lys, xlinks):
            xdist = []
```

```
        for xl in xlinks.values.tolist():

            A = lys[(lys['chainID'] == xl[0]) & (lys['resSeq'] == xl[1])]
            B = lys[(lys['chainID'] == xl[2]) & (lys['resSeq'] == xl[3])]

            A = A[['x', 'y', 'z']].values[0]
            B = B[['x', 'y', 'z']].values[0]

            xdist.append([A,B])

        xdist = np.array(xdist)

        return np.linalg.norm(xdist[:,0]-xdist[:,1], axis=1)
```

B.3 Provide Values for Atomic Radii and Masses

```
In [8]: radii = {
                'H' : 1.100,
                'C' : 1.872,
                'N' : 1.507,
                'O' : 1.400,
                'S' : 1.848}

        masses = {
                'H':1.00794,
                'C':12.0107,
                'N':14.0067,
                'O':15.9994,
                'Na':22.98976928,
                'Mg':24.3050,
                'P':30.973762,
                'S':32.065,
                'Cl':35.453,
                'K':39.0983,
                'Ca':40.078}
```

B.4 Load Data

Load atoms as dataframe. The first five lines are shown.

```
In [9]: atoms = pdb_to_df('4i6j.pdb')
```

```
Out[9]:   Record  serial name altLoc Resn chainID  resSeq iCode       x      y  \
        1   ATOM       1    N         ALA       A      21       -24.735  7.280
        2   ATOM       2   CA         ALA       A      21       -25.872  6.526
        3   ATOM       3    C         ALA       A      21       -25.496  5.716
        4   ATOM       4    O         ALA       A      21       -24.760  4.735
```

```
5   ATOM       5   CB         ALA       A      21       -26.447  5.616

        z  occupancy  tempFactor element charge
1 -10.679          1       71.10       N
2 -10.161          1       72.66       C
3  -8.916          1       68.57       C
4  -9.000          1       72.94       O
5 -11.248          1       61.64       C
```

Load cross-links as a data frame. As described above, cross-links involving residues missing from the PDB file are excluded. Residue numbering has been fixed using the `ResnConvert` command. The resulting data frame contains the chain and residue number for each cross-linked residue, with each row representing a pair of cross-linked residues.

```
In [10]: xlinkfiles = ['xlinks-proteins-run-78-low-salt.txt',
                       'xlinks-proteins-run-100-high-salt.txt']

         prots = {'mCRY2-1-544-mouse': 'A', 'Fbxl3-human': 'B', 'Skp1dd-human': 'C'}

         xlinks = read_xlinks(xlinkfiles, prots)
         xlinks.head()
```

```
Out[10]:   PROTEIN 1  POSITION PROTEIN 2  POSITION.1
         0         C       128         C         137
         1         C       128         C         142
         2         C       128         C         155
         3         C       128         B         116
         4         C       130         C         130
```

B.5 Set Up Model

Generate list of xlinks which will be the names of the columns.

```
In [11]: xlink_labels = ['%s/%03d -- %s/%03d' % tuple(x) for x in xlinks.values.tolist()]
```

List the domains as they will be called in the rest of the model and the summary file.

```
In [12]: domains = ["CRY2", "FBXL3", "SKP1", "SKPFBXL"]
```

Dictionaries of pivot points and interface logics. For each domain, the `pivot_points` dictionary will provide the Cartesian coordinates about which to pivot the structure. The `mobiles` dictionary contains the text strings that are used to query the `atoms` data frame to collect the atoms that are mobile during the pivot.

```
In [13]: pivot_points = {'CRY2': np.array([-52.672, -16.232, -6.874]),
                         'FBXL3': np.array([-88.486, -56.556, 2.886]),
                         'SKP1': np.array([-76.639,-11.300,-5.773]),
                         'SKPFBXL': np.array([-90.714,-20.252,4.378])}

In [14]: mobiles = {'CRY2': 'serial < 2386',
                    'FBXL3': 'serial > 7237 | 4081 < serial < 5566',
                    'SKP1': '7237 < serial < 7995',
                    'SKPFBXL': '4080 < serial < 4396 | 7237 < serial < 8283'}
```

Generate the array of angles for use in the `for` loop. To sample the full conformational space, the structure is pivoted by 2π radians about two orthogonal axes and π radians about a third. Note in that the final model $n = 30$ and as such the angles will increment by $\frac{\pi}{15}$, yielding the following 13500×3 array.

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 2 \\ . & . & . \\ . & . & . \\ . & . & . \\ 29 & 29 & 12 \\ 29 & 29 & 13 \\ 29 & 29 & 14 \end{bmatrix} \times \frac{2\pi}{n} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{\pi}{15} \\ 0 & 0 & \frac{2\pi}{15} \\ . & . & . \\ . & . & . \\ . & . & . \\ \frac{29\pi}{15} & \frac{29\pi}{15} & \frac{4\pi}{5} \\ \frac{29\pi}{15} & \frac{29\pi}{15} & \frac{13\pi}{15} \\ \frac{29\pi}{15} & \frac{29\pi}{15} & \frac{14\pi}{15} \end{bmatrix}$$

```
In [15]: n = 30

         angles = np.mgrid[0:n,0:n,0:n/2]
         angles = angles.reshape(3, n**3 / 2)
         angles = np.transpose(angles)
         angles = angles * n**-1 * 2 * np.pi
         angles

Out[15]: array([[ 0.        ,  0.        ,  0.        ],
                [ 0.        ,  0.        ,  0.20943951],
                [ 0.        ,  0.        ,  0.41887902],
                ...,
                [ 6.0737458 ,  6.0737458 ,  2.51327412],
                [ 6.0737458 ,  6.0737458 ,  2.72271363],
                [ 6.0737458 ,  6.0737458 ,  2.93215314]])
```

B.6 Run Model with nested `for` loop

```
In [16]: summary_data = []

         for domain in domains:
             pivot_point = pivot_points[domain]
             mobile = atoms.query(mobiles[domain])
             static = atoms[np.logical_not(atoms['serial'].isin(mobile['serial']))]

             mobile_ = mobile[['serial', 'x', 'y', 'z']].values
             static_ = static[['serial', 'x', 'y', 'z']].values

             m_radii = np.array([radii[x] for x in mobile['element']])
             s_radii = np.array([radii[x] for x in static['element']])

             min_dist = np.add.outer(m_radii, s_radii)

             for i in range(n**3 / 2):
                 phi, theta, psi = angles[i]
                 mobileR = RotatePoint(mobile_[:,1:4], pivot_point, phi, theta, psi)

                 rotAtoms = atoms.copy()
                 rotAtoms.loc[rotAtoms['serial'].isin(mobile_[:,0]), ['x', 'y', 'z']] = mobileR

                 # Clashes
                 dist_matrix = dist.cdist(mobile_[:,1:4], static_[:,1:4], metric='euclidean')
                 clash = dist_matrix - min_dist
                 clashes = (clash < 0).any(axis=0).sum() + (clash < 0).any(axis=1).sum()

                 # CCS Calculations
                 CCS_coords = rotAtoms[['x', 'y', 'z', 'element']]
                 CCS_coords.loc[:,'element'] = [masses[x] for x in CCS_coords['element']]

                 W = CalcCCS(CCS_coords.values, 16)
                 W = 0.84*W[:,0] + 0.22*W[:,1]
                 W_mean = np.mean(W)
                 W_stdev = np.std(W)
                 W_CI = W_stdev / np.sqrt(np.size(W))

                 # Measure the Cross-Link Distances
                 lysines = rotAtoms[(rotAtoms['Resn'] == 'LYS') & (rotAtoms['name'] == 'CA')]
                 valid_xlinks = EvalXlinks(lysines, xlinks)
                 data = [i, phi, theta, psi, clashes,
                           W_mean, W_stdev, W_CI] + valid_xlinks.tolist()
                 summary_data.append(data)

         headers = ['index', 'phi', 'theta', 'psi', 'clashes', 'CCS',
                             'Standard Deviation', 'Confidence Interval'] + xlink_labels

         summary_data = pd.DataFrame(summary_data, columns=headers)
```

```
summary_data.to_csv('summary.csv', float_format='%.4f')
```

B.7 Applicability and Conclusions

The model as described in this appendix has been tailored for the CRY2–FBXL3–SKP1 system and as such is specific to a single protein complex. However, it establishes a new framework for generating and evaluating conformationally flexible protein complexes. The power in this approach comes from making coarse estimates of flexibility using the GNM (Appendix A) and then performing a very basic geometric rotation of the structure. With all-atom models, atomic clashes can be determined to eliminate overlapping structures. Further analysis of the generated structures should be thought of as being modular, and can be adapted based on the experimental data available, computational resources, and specific questions.

Appendix C

A SCORING FUNCTION FOR EVALUATING CANDIDATE STRUCTURES OF THE CRY2–FBXL3–SKP1 EXTENDED CONFORMER

C.1 Introduction

The Python code used to implement the scoring function and analyze its results is shown herein. The software is initialized by importing the following libraries

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import matplotlib.gridspec as gridspec
        import seaborn as sns
        from mpl_toolkits.mplot3d import Axes3D

        %matplotlib inline
```

C.2 Definition of Scoring Functions

The values for each score term will be normalized to be between 0 and 1 using the following function. Normalization at this step ensures that the final scores will all fall between 0 and 1 and that the contributions from each score term are on a similar scale.

```
In [2]: def norm(x):
            x = x - x.min()
            x = x / x.max()

            return x
```

Briefly, the first score term S_Ω calculates the ratio of a structure's calculated collision cross section Ω_{calc} to the average of the experimentally obtained collision cross section values for the extended conformer $\langle\Omega_{exp}\rangle$. Collision cross sections of each structure were calculated

using a linear combination of the PA and EHSS as described in Chapter 3. To ensure strong statistical confidence, 16 different values were calculated for each structure, such that the uncertainty as defined by the 95% confidence interval would be less than 3.0% for any given structure. The max error among all values in the dataset was 2.1%, well within the 3.0% tolerance for drift tube measurements. The largest increases in Ω_{calc} were seen from bending at the FBXL3 pivot point, with the largest structure generated having a collision cross section of 73.6 nm^2, a 13.4% increase in Ω compared with the value calculated for the PDB structure.

$$S_{\Omega} = \left(\frac{\Omega_{calc}}{\langle \Omega_{exp} \rangle} \right) \tag{C.1}$$

The function to calculate S_{Ω} in Python is implemented as follows

```
In [3]: def ccs_score(CCS, CCS_avg):
            CCS = CCS.astype('float')
            CCS_avg = float(CCS_avg)
            ccs_score = CCS / CCS_avg
            ccs_score = norm(ccs_score)

            return ccs_score
```

The cross-link score S_{xlink} indicates how well a structure satisfies the hard cutoff of 30.0 Å that was used to evaluate cross-links in Hoopman et al. [94]. Rather than using a hard cutoff, however, we chose a sigmoidal function centered at 30.0 Å. As there are two conformational populations, it is not necessary for any given structure to satisfy all the cross-links. Rather, a structure should be scored well if the distance between a pair of cross-linked residues is reduced compared to the published crystal structure. This was evaluated by calculating the score for a cross-link in the original structure and the score for that corresponding cross-link in the rotated structure. If the score improves, then the difference between the score for the PDB structure and the rotated structure is calculated. This is repeated for every cross-link. These differences, representing the magnitude of improvement for each shortened cross-link distance, are then summed to determine S_{xlink} of that structure.

In doing so, cross-links that do not improve in the rotated structures do not penalize the score. Further, by centering the sigmoidal function at 30.0 Å, changes in distance from above the original cutoff to below are weighted the most heavily.

$$S_{xlink} = \sum \Delta \left(\frac{1}{1 + e^{0.5(z_i - 30.0)}} \right) \tag{C.2}$$

The function to calculate S_{xlink} in Python is implemented as follows

```
In [4]: def xlink_score(xl_dist_pdb, xl_dist_rot, mu):
            sigmoid = lambda z: (1. / (1 + np.exp(0.5 * z)))

            xl_dist_pdb = xl_dist_pdb - mu
            xl_dist_rot = xl_dist_rot - mu

            pdb_score = sigmoid(xl_dist_pdb)
            rot_score = sigmoid(xl_dist_rot)

            score = (rot_score > pdb_score) * (rot_score - pdb_score)
            score = np.sum(score, axis=1)
            score = norm(score)

            return score
```

The third score calculated considers the steric clashing of the generated structure (S_{clash}). The model directly outputs the number of atoms that are clashing, so S_{clash} is simply the fraction of non-clashing atoms. Structures with a high number of clashing atoms will therefore be penalized, whereas a structure having no clashes would have a score of 1.

$$S_{clash} = \left(\frac{n_{total} - n_{clash}}{n_{total}} \right) \tag{C.3}$$

The function to calculate S_{clash} in Python is implemented as follows

```
In [5]: def clash_score(clashes):
            clashes = clashes.astype('float')
            clash_score = (8279. - clashes) / 8279.
            clash_score = norm(clash_score)

            return clash_score
```

The product of these terms yields the scoring function S.

$$S = \left(\frac{\Omega_{calc}}{\langle\Omega_{exp}\rangle}\right) \times \left(\sum \Delta \frac{1}{1 + e^{0.5(z_i - \mu)}}\right) \times \left(\frac{n_{atoms} - n_{clashing}}{n_{atoms}}\right) \tag{C.4}$$

C.3 Load and Preprocess Model Data

The data file (generated by the model described in Appendix B) is loaded as a data frame. The first five rows are shown.

```
In [6]: data = pd.read_csv('cry2-fbxl3-skp1_model_data.csv', sep=',',
                                                  skipinitialspace=True, header=0)
```

Out[6]:

	index	Domain	phi	theta	psi	Clashes	CCS	Standard Deviation	\
0	0	CRY2	0	0	0.0000	81	6489.7437	50.4080	
1	1	CRY2	0	0	0.2094	42	6609.7537	58.1481	
2	2	CRY2	0	0	0.4189	39	6710.9322	76.5110	
3	3	CRY2	0	0	0.6283	24	6793.5307	60.1688	
4	4	CRY2	0	0	0.8378	16	6845.9053	67.6569	

	Confidence Interval	C/128 -- C/137	...	C/094 -- C/130	\
0	12.6020	14.18782	...	20.31807	
1	14.5370	14.18782	...	20.31807	
2	19.1278	14.18782	...	20.31807	
3	15.0422	14.18782	...	20.31807	
4	16.9142	14.18782	...	20.31807	

	C/128 -- B/092	B/092 -- B/178	B/092 -- B/190	B/140 -- B/178	\
0	21.417415	22.622827	35.969286	11.876339	
1	21.417415	22.622827	35.969286	11.876339	
2	21.417415	22.622827	35.969286	11.876339	
3	21.417415	22.622827	35.969286	11.876339	
4	21.417415	22.622827	35.969286	11.876339	

	B/140 -- B/190	B/140 -- B/201	B/172 -- B/190	B/201 -- B/248	\
0	18.305156	10.150376	14.393774	14.540409	
1	18.305156	10.150376	14.393774	14.540409	
2	18.305156	10.150376	14.393774	14.540409	
3	18.305156	10.150376	14.393774	14.540409	
4	18.305156	10.150376	14.393774	14.540409	

	A/107 -- A/169
0	50.450541
1	50.450541
2	50.450541
3	50.450541
4	50.450541

[5 rows x 78 columns]

Each row represents a generated structure. The second column (`Domain`) is the pivot point at which the structure was rotated. The next three columns, `phi`, `theta`, and `psi`, are the angles by which the structure was rotated. The sixth column is the number of atomic clashes. The seventh through ninth columns are Ω_{calc} values (the mean of 16 calculations is reported), the standard deviation of the Ω_{calc} values, and the confidence interval (defined as $\frac{\sigma}{N^2}$, where $N = 16$). The tenth through final columns are all cross-link distances. To subset these columns by name, the column names are extracted and stored in the variable `xlinks_names` as a list of strings. The cross-link values are then subset with the expression `data[xlinks_names]` and stored in the variable `xlinks`. Additionally, the cross-link distances for the original PDB structure is subset by selecting one of the four structures where φ, θ, and ψ are all equal to 0.0. The scores of these cross-link distances are the ones to which all the others will be compared to determine if the score has improved from the rotation.

```
In [7]: xlinks_names = list(data.columns.values)[9:]
        xlinks = data[xlinks_names]
        xlinks_pdb = xlinks[(data['phi'] == 0.0) &
                            (data['theta'] == 0.0) &
                            (data['psi'] == 0.0) &
                            (data['Domain'] == 'CRY2')]
```

The constants used in the scoring function, such as μ and $\langle\Omega_{exp}\rangle$, must be set before proceeding. The value for μ, which represents the cutoff distance for the cross-link term, is set to 30.0 Å. The average experimental Ω value for the extended conformer (`CCS_avg`) is set to 8014.818 Å^2. Finally, all the data are converted from data frame series to NumPy arrays before being passed to the scoring functions.

```
In [8]: mu = 30.0
        CCS_avg = 8014.818

        CCS_array = data['CCS'].values
        xlink_array = xlinks.values
        xlink_pdb_array = xlinks_pdb.values
        clashes_array = data['Clashes'].values
```

C.4 Evaluation of Scoring Function

A new empty data frame called `score_data` is constructed that will hold the scores for each structure. The columns for the domain and angles (`['Domain', 'phi', 'theta', 'psi']`) are subset from `data` and copied to `score_data`. Three new columns — `ccs term`, `xlink term`, and `clash term` — are added holding the values for each corresponding score term — S_Ω, S_{xlink}, and S_{clash} — as well as the product of these S in the column named `Score`.

```
In [9]: score_data = pd.DataFrame()

        for column in ['Domain', 'phi', 'theta', 'psi']:
            score_data[column] = data[column]

        score_data['ccs term'] = ccs_score(CCS_array, CCS_avg)
        score_data['xlink term'] = xlink_score(xlink_pdb_array, xlink_array, mu)
        score_data['clash term'] = clash_score(clashes_array)

        terms = ['ccs term', 'xlink term', 'clash term']
        score_data['Score'] = np.product(score_data[terms].values, axis=1)
```

Out[9]:

	Domain	phi	theta	psi	ccs term	xlink term	clash term	Score
0	CRY2	0	0	0.0000	0.595123	0.000000	0.982112	0.000000
1	CRY2	0	0	0.2094	0.650878	0.003979	0.991669	0.002568
2	CRY2	0	0	0.4189	0.697884	0.004732	0.992404	0.003277
3	CRY2	0	0	0.6283	0.736258	0.002605	0.996079	0.001910
4	CRY2	0	0	0.8378	0.760590	0.004705	0.998040	0.003572

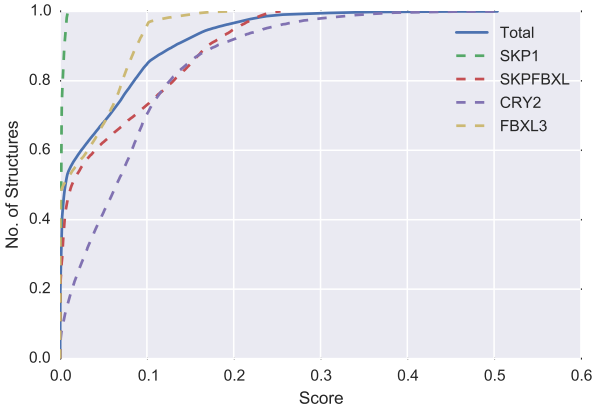
C.5 Statistics and Examination of Score Data

The cumulative distribution of scores is shown below. Most structures have scores of 0.2 or less.

```
In [10]: sorted_scores = np.sort(score_data['Score'].copy())
         y_vals = np.arange(54000).astype('float') / 54000.
         plt.plot(sorted_scores, y_vals, label='Total')

         for domain in domains:
             domain_score = score_data[score_data['Domain'] == domain]['Score']
             sorted_scores = np.sort(domain_score.copy())
             y_vals = np.arange(13500).astype('float') / 13500.
             plt.plot(sorted_scores, y_vals,'--', label=domain)

         plt.xlabel('Score')
```

```
        plt.ylabel('No. of Structures')
        plt.legend()
```

```
Out[10]: <matplotlib.legend.Legend at 0x1108cc810>
```



The max score in the dataset is 0.5039.

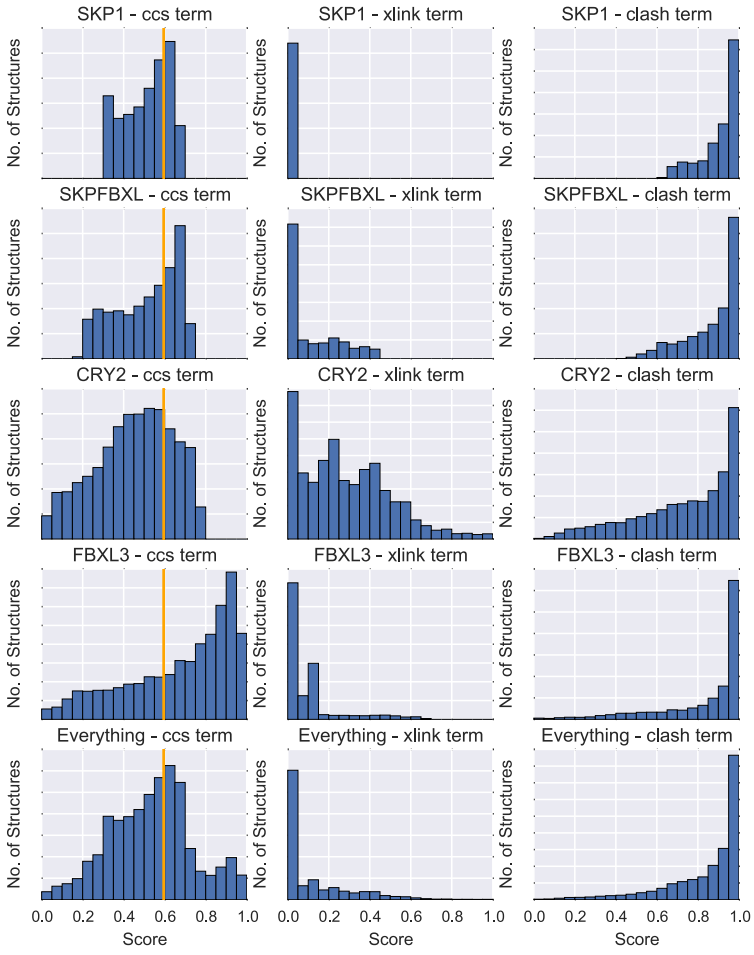
```
In [11]: score_data['Score'].max()
```

```
Out[11]: 0.50391431368692796
```

Shown below is a breakdown of the scores of the structures by pivot point and by term (equivalent to Figure 3.6).

```
In [12]: hist_bins = np.arange(21).astype('float') / 20.
         hist_bins = hist_bins.tolist()

         fig = plt.figure(figsize=(8, 10))

         for i, domain in enumerate(domains):
             plot_data = score_data[score_data['Domain'] == domain]
             for j, term in enumerate(terms):
                 ax = fig.add_subplot(5,3, i*3+j+1)
                 ax.hist(plot_data[term].values, bins=hist_bins, histtype='bar')
                 if term == 'ccs term':
                     ax.axvline(x=CCS_score_4i6j_mean, c='orange')
                 ax.set_title('%s - %s' % (domain, term))
                 ax.set_xlim(0.0, 1.0)
                 ax.set_xticklabels([])
                 ax.set_yticklabels([])
                 ax.set_ylabel('No. of Structures')
```

```
for j, term in enumerate(terms):
    ax = fig.add_subplot(5,3, 13+j)
    ax.hist(score_data[term].values, bins=hist_bins, histtype='bar')
    if term == 'ccs term':
        ax.axvline(x=CCS_score_4i6j_mean, c='orange')
    ax.set_title('Everything - %s' % term)
    ax.set_xlim(0.0, 1.0)
    ax.set_yticklabels([])
    ax.set_ylabel('No. of Structures')
    ax.set_xlabel('Score')
```



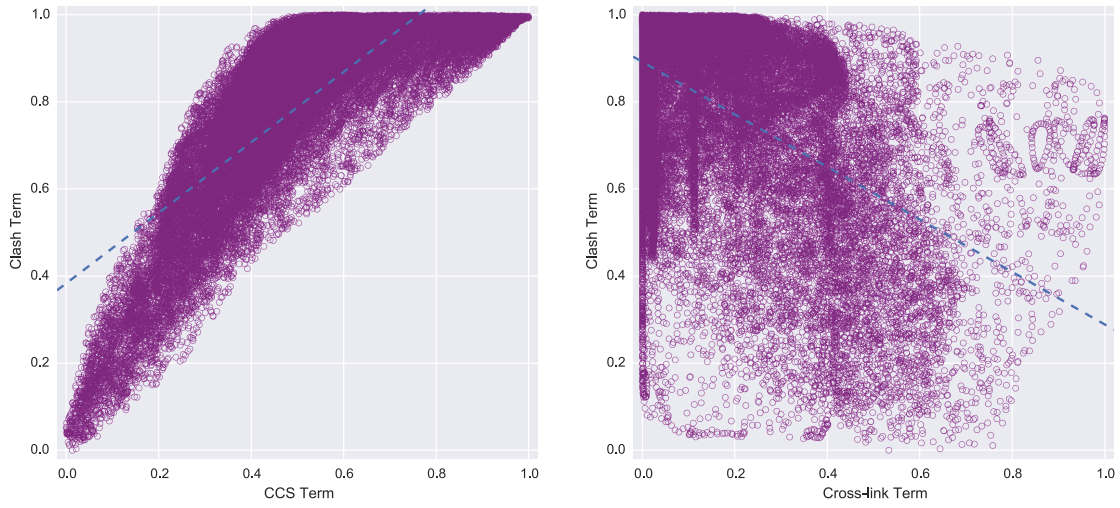
Below are plots showing the degree of correlation between S_{Ω} and S_{clash} as well as the correlation between S_{xlink} and S_{clash}. These results are discussed in Chapter 3.

```
In [13]: from scipy.stats import linregress
```

```
In [14]: fig = plt.figure()
```

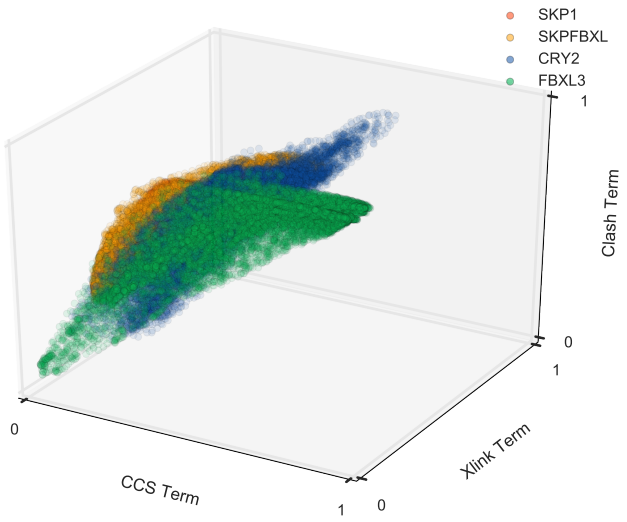
```
        ax0 = fig.add_subplot(121)
        ax0.scatter(score_data['ccs term'], score_data['clash term'],
                                            facecolor='none', edgecolor='purple')
        ax0.set_xlabel('CCS Term')
        ax0.set_ylabel('Clash Term')

        m, b, r, p, std_err = linregress(score_data['ccs term'], score_data['clash term'])
        print "Pearson's r (CCS vs Clash)\t", r

        x = np.linspace(-0.020,1.020,10)
        y = m * x + b
        ax0.plot(x,y, ls='--')
        ax0.set_xlim(-0.020, 1.020)
        ax0.set_ylim(-0.020, 1.020)

        ax1 = fig.add_subplot(122)
        ax1.scatter(score_data['xlink term'], score_data['clash term'],
                                            facecolor='none', edgecolor='purple')
        ax1.set_xlabel('Cross-link Term')
        ax1.set_ylabel('Clash Term')

        m, b, r, p, std_err = linregress(score_data['xlink term'], score_data['clash term'])
        print "Pearson's r (X-link vs Clash)\t", r

        x = np.linspace(-0.020,1.020,10)
        y = m * x + b
        ax1.plot(x,y, ls='--')
        ax1.set_xlim(-0.020, 1.020)
        ax1.set_ylim(-0.020, 1.020)
```

```
Pearson's r (CCS vs Clash)        0.796366019417
Pearson's r (X-link vs Clash)        -0.532074635277

Out[14]: (-0.02, 1.02)
```



Below is a scatter plot showing the contribution of each term. Each axis represents one score term.

```
In [15]: fig = plt.figure()
         ax = fig.add_subplot(111, projection='3d')
         colors = ['#FF3900', '#FF9E00', '#0E51A7', '#00B454']

         for j, domain in enumerate(domains):
             plot_data = score_data[score_data['Domain'] == domain][terms].values
             ax.scatter(*plot_data.T, c=colors[j], alpha=0.10, label=domain)

         ax.set_xlabel(u'CCS Term')
         ax.set_ylabel(u'Xlink Term')
         ax.set_zlabel(u'Clash Term')

         for set_ticks in [ax.set_xticks, ax.set_yticks, ax.set_zticks]:
             set_ticks([0.0, 1.0])

         for set_lim in [ax.set_xlim, ax.set_ylim, ax.set_zlim]:
             set_lim(0.0, 1.0)

         ax.legend()

Out[15]: <matplotlib.legend.Legend at 0x11348e7d0>
```

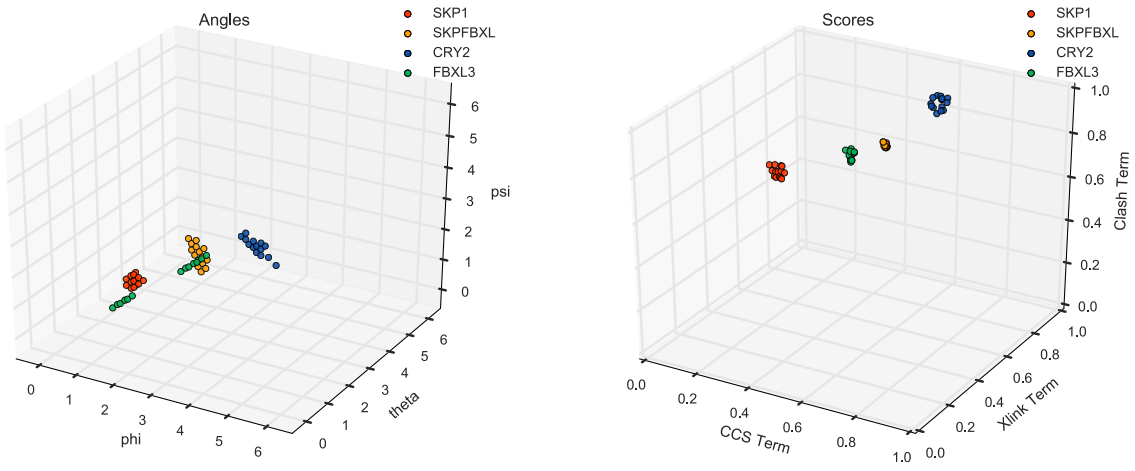


C.6 Results

Shown are the top 0.1% scoring structures for each pivot point. On the left, the structures are plotted in angle space to show that the top scoring structures for any given pivot point are geometrically similar to each other. On the right, they are shown plotted against the score terms as above.

```
In [16]: fig = plt.figure()
         colors = ['#FF3900', '#FF9E00', '#0E51A7', '#00B454']

         ax = fig.add_subplot(121, projection='3d')
         ax.set_title('Angles')
         for j, domain in enumerate(domains):
             plot_data = score_data[score_data['Domain'] == domain]
             plot_data = plot_data[plot_data['Score'] > plot_data['Score'].quantile(0.999)]
             plot_data = plot_data[['phi','theta','psi']].values
             ax.scatter(*plot_data.T, c=colors[j], alpha=0.90, label=domain)

         ax.set_xlabel(u'phi')
         ax.set_ylabel(u'theta')
         ax.set_zlabel(u'psi')
         ax.legend()

         for set_lim in [ax.set_xlim, ax.set_ylim, ax.set_zlim]:
             set_lim(-0.5, 6.5)


         ax = fig.add_subplot(122, projection='3d')
```

```
ax.set_title('Scores')
for j, domain in enumerate(domains):
    plot_data = score_data[score_data['Domain'] == domain]
    plot_data = plot_data[plot_data['Score'] > plot_data['Score'].quantile(0.999)]
    plot_data = plot_data[terms].values
    ax.scatter(*plot_data.T, c=colors[j], alpha=0.90, label=domain)

ax.set_xlabel(u'CCS Term')
ax.set_ylabel(u'Xlink Term')
ax.set_zlabel(u'Clash Term')
ax.legend()

for set_lim in [ax.set_xlim, ax.set_ylim, ax.set_zlim]:
    set_lim(0.0, 1.0)
```



The top scoring structure in the entire dataset can be queried as follows

```
In [17]: score_data[score_data['Score'] == score_data['Score'].max()]
```

```
Out[17]:       Domain     phi  theta     psi  ccs term  xlink term  clash term  \
         11554   CRY2  2.5133  5.236  0.8378  0.617771    0.904302     0.90202

                  Score
         11554  0.503914
```

From this it is clear that improvements to cross-link distances contributed the most to the high score of this structure. For comparison, the structure with the highest S_Ω can be queried as follows

```
In [18]: score_data[score_data['ccs term'] == score_data['ccs term'].max()]
```

```
Out[18]:       Domain     phi   theta     psi  ccs term  xlink term  clash term  \
         28031  FBXL3  0.2094  0.8378  2.3038         1     0.00009    0.997948

                  Score
         28031  0.00009
```

Thus it appears that pivoting the CRY2 subunit satisfies many cross-links, while pivoting the FBXL3 subunit results in the greatest increase in Ω_{calc}.

C.7 Applicability and Conclusions

Herein a scoring function has been described for analyzing a combination of cross-linking and ion mobility data for the CRY2–FBXL3–SKP1 complex, while eliminating overlapping structures. The structure of the scoring function should be revisited if new data were available to better inform this approach, or if these data were to be integrated into the model as a fourth term. However, it serves as a robust starting point for analyzing such datasets in a rapid and automatic way.

Appendix D

NATIVEFIT: NATIVE MASS SPECTRAL FITTING SOFTWARE

```
from __future__ import absolute_import, division, print_function

import time
import numpy as np
import pandas as pd
from scipy.interpolate import InterpolatedUnivariateSpline
from scipy.optimize import leastsq
from scipy.stats import norm

__version__ = '1.0.0'

class SpectrumFit(object):

    def __init__(self, fname, mz_range):
        self.mzLow, self.mzHigh, step = mz_range

        self.mz = np.arange(self.mzLow, self.mzHigh, step)
        self.y = np.zeros_like(self.mz)
        self.ys = np.ndarray([])

        ms = np.genfromtxt(fname).T
        ms = ms[:, (ms[0] >= self.mzLow) & (ms[0] <= self.mzHigh)]
        ms[1] = ms[1] - np.min(ms[1])
        ms[1] = ms[1] / np.max(ms[1])
        self.ms_new = InterpolatedUnivariateSpline(*ms)
        self.exp = self.ms_new(self.mz)

        data = {col:[] for col in self.columns}
        self.Complexes = pd.DataFrame(data,columns=self.columns,index=[])


    def add(self, name):
        if name in list(self.Complexes.index.values):
            print("Overwrote existing assembly.")
        else:
            print("Created a new assembly.")


    def plot(self, ax, fill=False):
        """ Plots a figure of the mass spectrum and the fit.
```

```
    Parameters
    ----------
    ax : object
        The subplot upon which the spectrum is plotted.
    fill : Optional[False]
        The simulated spectra are filled in.
    """
    colors = ['blue', 'green', 'red', 'teal']

    ax.plot(self.mz, self.exp, linewidth=1, color='k')
    if self.ys.size > 0:
        for i in np.arange(self.ys.shape[0]):
            ax.plot(self.mz, self.ys[i], linewidth=1, color=colors[i])
            if fill == True:
                ax.fill_between(self.mz, 0, self.ys[i], color=colors[i], alpha=0.3)

    ax.set_xlim(self.mzLow, self.mzHigh)
    ax.set_ylim(0.0, 1.1)


def extract_parameters(self):
    return self.Complexes[self.columns[:]].values


def calc_spectrum(self):
    self.calc_parameters_spectra(self.extract_parameters())


def score(self):
    return np.sum(np.square(self.y-self.exp))

def peaks_fit(self, peaks, z_range=(1,100), plot=False):
    peaks = np.array(peaks)
    possible_z = np.arange(*z_range)

    z = np.add.outer(possible_z, np.arange(peaks.size)[::-1])
    masses = np.array(peaks) * z - z * 1.0078

    candidates = pd.DataFrame()
    candidates['mass'] = np.average(masses, axis=1)
    candidates['z_avg guess'] = np.average(z, axis=1)
    candidates['score'] = np.std(masses, axis=1)

    # Sort by score and reset index
    candidates.sort('score', inplace=True)
    candidates.index = np.arange(possible_z.size)

    if plot == True:
        xmin = candidates.loc[0:10,:]['mass'].min() * 0.9
```

```
            xmax = candidates.loc[0:10,:]['mass'].max() * 1.1
            ymax = candidates.loc[0:10,:]['score'].max() * 1.2
            candidates.plot('mass', 'score', 'scatter', xlim=(xmin, xmax), ylim=(0, ymax))

        return candidates


class native(SpectrumFit):

    def __init__(self, fname, mz_range):
        self.thresh = 1e-5
        self.mres = 850
        self.zw = 4.0

        self.columns = ['mass','z','Rp','width', 'pop']
        super(native, self).__init__(fname, mz_range)


    def plot(self, ax, fill=False):
        self.calc_spectrum()
        super(native, self).plot(ax, fill)


    def add(self, name, mass, zavg, mres=450, zwidth=1.5, pop=60):
        super(native, self).add(name)
        self.Complexes.loc[name,:] = [mass, zavg, mres, zwidth, pop]


    def remove(self, name):
        self.Complexes = self.Complexes.drop(name, axis=0)


    def update_parameters(self,data):
        self.Complexes[self.columns[:]] = data


    def calc_parameters_spectra(self,data):
        num_assemblies = data.shape[0]

        ys = np.zeros([num_assemblies, self.mz.size])

        for i in np.arange(num_assemblies):
            zval = np.arange(1,500)
            zdist = data[i,4]*norm.pdf(zval,data[i,1],data[i,3])

            keepers = np.greater(zdist,self.thresh)
            mz_centroids = 1+data[i,0]/zval[np.nonzero(keepers)]
            zdist = zdist[np.nonzero(keepers)]
            res = mz_centroids/(2.355*data[i,2])
```

```
            for j in np.arange(mz_centroids.size):
                ys[i] += zdist[j]*norm.pdf(self.mz, mz_centroids[j], res[j])

        self.ys = ys
        self.y = np.sum(ys, axis=0)
        return self.y


    def optimize(self, mass=True, zavg=True, mres=False, zwidth=True, pop=True):
        mask_vals = np.array([mass, zavg, mres, zwidth, pop])
        parameters = self.extract_parameters()
        print(parameters)

        def eval_config(p,mask):
            indexices = np.transpose(np.nonzero(mask))
            for i in range(len(p)):
                parameters[indexices[i,0],indexices[i,1]]=p[i]
            return self.calc_parameters_spectra(parameters)

        t0 = time.time()

        mask = np.zeros_like(self.Complexes)
        for i, val in enumerate(mask_vals):
            mask[0:parameters.shape[0], i] = val
        errfunc = lambda p,mask: self.exp-eval_config(p,mask)
        p0 = []
        for item in np.transpose(np.nonzero(mask)):
            p0.append(parameters[item[0],item[1]])

        opt = leastsq(errfunc, p0, args=(mask))

        for x in np.reshape(opt[0], (parameters.shape[0],-1)):
            a = ['{:.8e}'.format(y) for y in x]
            print('[%s]' % ', '.join(map(str, a)))

        t_elapsed = time.time() - t0
        print("Score: %.6f\nTime: %.6f\n" % (self.score(), t_elapsed))

        indexices = np.transpose(np.nonzero(mask))
        p = opt[0]
        for i in range(len(p)):
            parameters[indexices[i,0],indexices[i,1]]=p[i]

        return parameters


class captr(SpectrumFit):

    def __init__(self, fname, mz_range):
        self.columns = ['mass','z_min','z_max','Rp']
```

```
        self.num_assemblies = 0
        self.print_peaks = False
        self.print_Rp = False
        super(captr, self).__init__(fname, mz_range)


    def add(self, name, mass, z_range, Rp=450):
        mass = float(mass)
        zLow, zHigh = z_range
        zHigh += 1
        self.num_assemblies += 1

        super(captr, self).add(name)
        self.Complexes.loc[name,:] = [mass, zLow, zHigh, Rp]


    def optimize(self, print_Rp=False, print_peaks=False):
        ys = np.zeros((self.num_assemblies, self.mz.size))
        Complexes = self.Complexes.values

        Gaussian = lambda t,p : p[0]*np.exp(-(t-p[1])**2/(2*p[2]**2))
        errfunc = lambda p, X, Y: Gaussian(X,p)-Y

        for i in range(self.num_assemblies):
            mass, zLow, zHigh, Rp = Complexes[i]    #########

            peaks = [(mass + z) / float(z) for z in np.arange(zLow, zHigh)]
            peaks = np.array(peaks).astype('float')[::-1]

            Gaussians = np.zeros((peaks.size, self.mz.size))
            opt_peaks = np.zeros_like(peaks)
            opt_sigma = np.zeros_like(peaks)

            # Starting parameters for the Guassian Fit [Intensity, mean, sigma]
            p = np.zeros((peaks.shape[0], 3))
            p[:,0] = self.ms_new(peaks)
            p[:,1] = peaks
            p[:,2] = peaks / (Rp * 2.35482004503)

            for j in range(peaks.size):
                start = p[j,1] - 2 * p[j,2]
                end = p[j,1] + 2 * p[j,2]
                x = np.arange(start,end, 1)
                y = self.ms_new(x)

                opt, success = leastsq(errfunc,p[j],args=(x,y))
                opt_peaks[j] = opt[1]
                opt_sigma[j] = opt[2]
                Gaussians[j] = Gaussian(self.mz, opt)
```

```
        if print_peaks == True:
            print(list(opt_peaks))
            print('\n')

        if print_Rp == True:
            opt_Rp = opt_peaks / (opt_sigma * 2.35482004503)
            print(list(opt_Rp))
            print('\n')

        ys[i] = Gaussians.sum(axis=0)

    self.ys = ys
```

VITA

Samuel T. Marionni is originally from Greenbelt, MD, where he graduated from Eleanor Roosevelt High School in 2005 as a National Merit Scholar. He attended college at the University of Maryland in College Park, MD, receiving a Bachelor of Science in Chemistry in 2009. Sam began his doctoral program at the University of Washington in the fall of 2009. Initially under the supervision of Professor Gojko Lalic, Samuel researched copper catalysts for the development of a stereoselective synthesis of trisubstituted allenes before beginning his work with Professor Matthew F. Bush using native ion mobility-mass spectrometry to investigate biological structures. Sam currently lives in Seattle, WA and obtained his Ph.D. from the University of Washington in 2015.